

**PERANCANGAN DAN SIMULASI SISTEM KENDALI *ADAPTIVE* PID
PADA MOTOR *BRUSHLESS DIRECT CURRENT* (BLDC)
MENGUNAKAN METODE *MODEL REFERENCE ADAPTIVE*
CONTROLLER (MRAC)**

(Proposal Tugas Akhir)

Oleh :

ZAHWA DINDA AULIA

NPM. 1915031031



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

DAFTAR ISI

Halaman

HALAMAN JUDUL	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Hipotesis	4
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.2 Motor <i>Brushless</i> DC (BLDC).....	7
2.2.1 Prinsip Kerja Motor BLDC.....	8
2.2.2 Sensor Posisi	8
2.2.3 Pemodelan Matematika Motor BLDC	9
2.3 Sistem Kendali	14
2.3.1 Kalang Terbuka (<i>Open Loop</i>)	15
2.3.2 Kalang Tertutup (<i>Close Loop</i>)	15
2.3.3 Karakteristik Respon Sistem.....	16
2.4 Sistem Kendali <i>Proportional Integral Derivative</i> (PID)	17
2.5 Sistem Kendali Model <i>Reference Adaptive Controller</i> (MRAC)	19
2.5.1 Aturan <i>Massachusetts Institute of Technology</i> (MIT)	20
2.6 <i>Matrix Laboratory</i> (MATLAB)	20
BAB III METODE PENELITIAN	24
3.1 Waktu dan Tempat	24

3.2 Alat dan Bahan.....	24
3.3 Diagram Alir Perancangan Simulasi Sistem Kendali	26
3.4 Parameter Motor BLDC	28
3.4.1 Menentukan Pemodelan Matematika Motor BLDC.....	29
3.4.2 Merancang Sistem Kendali MRAC-PID	30
3.5 Merancang <i>Graphical User Interface</i> (GUI).....	30

DAFTAR PUSTAKA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini teknologi berkembang sangat pesat di berbagai bidang kehidupan seperti bidang industri, informasi, transportasi dan bidang-bidang lainnya. Salah satu contoh perkembangan teknologi yang ada saat ini adalah perkembangan motor listrik. Motor listrik adalah suatu perangkat yang dapat merubah energi listrik menjadi energi mekanik [4]. Salah satu motor listrik yang banyak digunakan saat ini yaitu Motor *Brushless* DC (BLDC).

Motor BLDC merupakan salah satu jenis motor listrik arus searah yang dirancang tanpa menggunakan sikat. Motor BLDC sudah banyak digunakan di industri seperti industri otomotif, konsumsi, kesehatan, otomasi industri dan instrumentasi [4]. Pada motor ini tidak menggunakan sikat, medan magnet pada stator dengan medan magnet pada rotor akan berputar dengan kecepatan yang sama dan dikendalikan secara elektronik sehingga mempunyai keuntungan seperti rasio inersia/torsi tinggi efisiensi tinggi, jangkauan pengaturan kecepatan yang besar dan rendahnya *electro-magnetic interference* (EMI) [4].

Meskipun memiliki banyak keunggulan, motor BLDC juga memiliki beberapa kekurangan antara lain yaitu kurangnya kestabilan kecepatan motor saat diberikan sebuah beban sedemikian rupa, hal itu menyebabkan kecepatan motor BLDC kurang stabil. Mengingat motor BLDC sudah banyak diaplikasikan pada berbagai bidang industri dan juga otomotif maka performa dari motor ini harus optimal. Untuk memastikan motor BLDC tetap dalam keadaan optimal diperlukan sistem rekayasa kendali sebagai pengontrol sistem.

Sistem kendali sudah banyak diaplikasikan pada peralatan elektronik, termasuk motor BLDC. Salah satu sistem kendali yang telah diaplikasikan yaitu sistem kendali kecepatan motor dengan kendali *Proportional Integral Derivative* (PID) tunggal. Sistem kendali ini hanya mampu bekerja untuk kondisi yang bersifat linier. Sehingga sistem kendali PID tunggal ini belum mampu menghasilkan keluaran yang cukup baik apabila terjadi perubahan *setpoint* [7].

Berdasarkan permasalahan diatas, maka penulis melakukan sebuah penelitian untuk merancang sebuah sistem kendali PID dalam mengendalikan kecepatan motor BLDC yang bersifat adaptif yang dapat menghasilkan keluaran kecepatan putar motor BLDC sesuai dengan yang diharapkan walaupun terjadi perubahan *setpoint*.

Terdapat beberapa penelitian mengenai sistem kendali PID pada motor BLDC, diantaranya adalah :

“Pengaturan Kecepatan Motor *Brushless* DC Menggunakan Kontroler *Fuzzy* Berbasis *Linear Quadratic Regulator*” oleh Fairuzza Dinansyar tahun 2016. Pada penelitian tersebut menggunakan kontroler *fuzzy* LQR untuk mengatur kecepatan motor BLDC agar respon yang dihasilkan sama untuk pembebanan yang bervariasi sesuai dengan spesifikasi yang diinginkan [11].

“*Speed Control of BLDC Motor using Adaptive Fuzzy PID Controller*” oleh R. Kandiban dan R. Arulmozhiyal tahun 2012. Dalam penelitian tersebut ingin membuktikan apakah benar bahwa kendali *Adaptive Fuzzy PID* dapat memberikan performa yang lebih baik dalam mengontrol kecepatan motor BLDC [1].

“Implementasi dan Analisis Kendali Kecepatan Motor BLDC 1 kW Menggunakan Algoritma PID” oleh Muchlis Dwi Ardiansyah dan Fatkhur Rohman tahun 2019. Pada penelitian tersebut pengambilan data dengan cara memasukkan nilai parameter secara *trial and error* dimana motor BLDC digabungkan dengan generator yang diberi beban lampu yang divariasikan [2].

“Pengendalian Kecepatan Motor *Brushless* DC (BLDC) Menggunakan Metode Logika Fuzzy” oleh Agung Dwi Yulianta, Sasongko Pramono Hadi dan Suharyanto pada tahun 2015. Dalam penelitian tersebut menggunakan metod logika *fuzzy* sebagai metode pengendalian kecepatan motor. [3].

Perbedaan penelitian ini dengan penelitian sebelumnya terletak pada metode yang digunakan. Metode yang digunakan adalah *Model Reference Adaptive Controller* (MRAC) untuk memodelkan sistem kendali dan mengontrol sistem kecepatan pada motor BLDC dan jenis PID yang digunakan adalah jenis *adaptive* PID. Perancangan ini berperan sangat penting guna meminimalisir slip atau gangguan, sehingga sistem kecepatan motor BLDC memiliki tingkat kestabilan yang baik.

1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah sebagai berikut.

1. Bagaimana cara pemodelan sistem kendali *adaptive* PID pada motor BLDC.
2. Bagaimana pengaruh sistem kendali *adaptive* PID terhadap kestabilan Motor BLDC.

1.3 Batasan Masalah

Adapun batasan masalah yang digunakan pada penelitian ini adalah sebagai berikut

1. Metode yang digunakan untuk penalaan parameter PID adalah *adaptive* PID.
2. *Plant* yang digunakan Motor BLDC.
3. Simulasi pengendalian menggunakan *software Simulink* MATLAB.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut.

1. Dapat memodelkan simulasi sistem kendali *adaptive* PID untuk kestabilan kecepatan Motor BLDC.
2. Mengetahui pengaruh sistem kendali *adaptive* PID terhadap kestabilan Motor BLDC.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah dapat mengetahui pengaruh sistem kendali *adaptive* PID pada sistem kecepatan motor BLDC sehingga dapat sistem kendali ini dapat digunakan derbagai peralatan yang menggunakan motor BLDC agar penggunaanya lebih efektif.

1.6 Hipotesis

Sistem kendali *adaptive* PID yang dirancang dan disimulasikan pada motor BLDC ini dapat mengontrol dan menjaga kestabilan kecepatan serta meminimalisir gangguan yang terjadi.

1.7 Sistematika Penulisan

Adapun sistematika penulisan Laporan Tugas Akhir ini adalah sebagai berikut.

BAB I PENDAHULUAN

Pada pendahuluan terdapat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, hipotesis dan sistematika penulisan.

BAB II TINJUAN PUSTAKA

Pada tinjauan pustaka berisi tentang teori-teori yang mendasari penelitian ini.

BAB III METODE PENELITIAN

Pada metode penilitian terdapat waktu dan tempat penelitian, alat dan bahan yang digunakan, metode yang akan digunakan, serta diagram alir penelitian.

BAB IV HASIL DAN PEMBAHASAN

Pada hasil dan pembahasan terdapat hasil penilitian serta analisis hasil penilitian yang diperoleh.

BAB V PENUTUP

Pada penutup terdapat rangkuman akhir atau kesimpulan yang diperoleh dari hasil dan pembahasan serta saran untuk pengembangan penelitian ini lebih lanjut.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Terdapat beberapa penelitian mengenai sistem kendali motor BLDC, diantaranya adalah:

“Implementasi dan Analisis Kendali Kecepatan Motor BLDC 1 kW Menggunakan Algoritma PID” oleh Muchlis Dwi Ardiansyah dan Fatkhur Rohman pada tahun 2019. Pada penelitian ini merancang sistem kendali motor BLDC menggunakan sistem kendali PID untuk mendapatkan nilai *error steady state* terkecil pada variasi kecepatan dan motor BLDC dengan menggunakan metode *trial error* melalui simulasi Simulink MATLAB. Hasil yang diperoleh dari penelitian ini yaitu nilai $K_p = 0,8$; $K_i = 6$ dan $K_d = 0,02$. Dengan nilai parameter tersebut motor BLDC dapat mempertahankan nilai *set point* dengan kestabilan yang tinggi (*error steady state* rendah) [2].

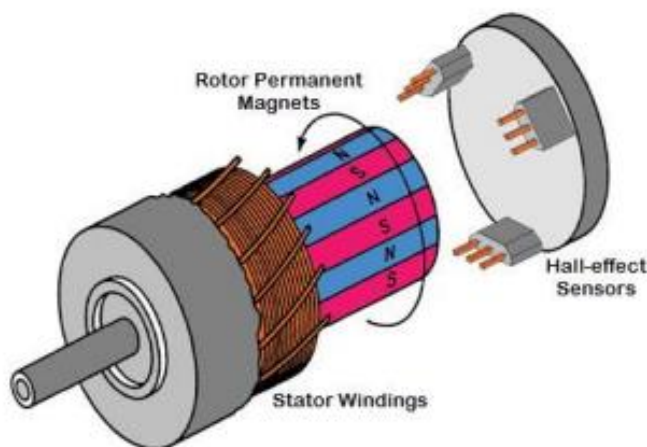
“Pengendalian Kecepatan Motor *Brushless* DC (BLDC) Menggunakan Logika Fuzzy” oleh Agung Dwi Yulianta, dkk pada tahun 2015. Pada penelitian ini pengendalian motor BLDC yang akan diuji menggunakan *fuzzy logic* dengan simulasi kecepatan keluaran berdasar referensi pengendalian fuzzy sehingga diperoleh hasil pengendalian yang optimal. Defuzzifikasi yang digunakan yaitu COA/centroid, bisektor, MOM, LOM, dan SOM. .Pada penelitian ini dihasilkan bahwa metode defuzzifikasi yang mampu mengikuti *setting* kecepatan yang diberikan adalah metode COA [3].

“Analisis Performa PID Pada Sistem Kontrol Kecepatan Motor BLDC” oleh Muhammada Abdhul Aziz pada tahun 2021. Pada penelitian ini menggunakan

metode tuning *Ziegler-Nichols* dengan *software* MATLAB. Dari penelitian tersebut dihasilkan parameter PID dengan nilai K_p 0.6996, nilai K_i 2.3320, dan nilai K_d 0.0525. Kemudian, hasil dari respon tanggapan sistem dibandingkan dengan hasil performa yang paling baik adalah kendali PI [4].

2.2 Motor *Brushless* DC

Motor *Brushless Direct Current* (BLDC) merupakan salah satu jenis motor DC yang merupakan motor sinkron yang tidak memiliki sikat. Motor BLDC ini memiliki stator yang berupa lilitan kawat yang digunakan sebagai medan magnet, rotor yang berupa magnet permanen serta memiliki 3 buah sensor *Hall* yang berfungsi untuk mendeteksi medan magnet dan menghasilkan tegangan yang proporsional.



Gambar 2.1 Konstruksi Motor BLDC

Motor BLDC memiliki kelebihan dibandingkan dengan motor DC konvensional lainnya. Kelebihan ini yang membuat motor BLDC banyak digunakan di bidang industri dan juga otomotif. Kelebihan dari motor BLDC yaitu :

1. Komutasinya dilakukan secara elektronik.
2. Jika dibandingkan dengan motor DC yang memiliki sikat, motor BLDC ini memiliki rentang kecepatan dan efisiensi yang lebih tinggi.
3. Umur operasi yang panjang dan saat beroperasi tidak ada suara atau tanpa suara.

2.2.1 Prinsip Kerja Motor BLDC

Prinsip kerja dari motor BLDC ini menggunakan prinsip elektromagnetik, gaya tarik menarik antara magnet permanen pada rotor dan electromagnet pada stator. Untuk membuat motor BLDC berputar atau bergerak, pada lilitan stator harus diberikan energi secara berurutan. Pada saat kutub magnet rotor berada di dekat sensor Hall, maka sensor ini akan memberikan sinyal tinggi atau rendah, yang menunjukkan kutub N atau S lewat di dekat sensor. Berdasarkan kombinasi dari tiga sinyal sensor Hall ini maka urutan pergantian yang tepat dapat ditentukan [4].

Saat motor berputar, magnet permanen pada rotor akan bergerak melewati kumparan pada stator yang kemudian akan menginduksi potensial listrik pada kumparan tersebut dan terjadilah BEMF atau *Back electromotive force*. BEMF ini berbanding lurus dengan kecepatan motor [6].

2.2.2 Sensor Posisi Motor BLDC

Sistem komutasi pada motor BLDC berbeda dengan sistem komutasi pada motor DC konvensional. Pada motor BLDC sistem komutasi dilakukan secara elektronik. Oleh sebab itu, maka diperlukan sensor yang dapat memberikan informasi secara presisi dan akurat kepada kontroler untuk mengatur lilitan pada stator manakah yang akan di aliri listrik. Pengaturan ini berfungsi agar kecepatan dan torsi pada motor tetap konstan [12]. Apabila perubahan komutasi yang dilakukan tidak sesuai atau tidak tepat maka akan terjadi slip pada motor BLDC, jika terjadi slip maka kecepatan dan torsi pada motor BLDC ini tidak konstan dan memungkinkan membuat motor berhenti berputar.

Kecepatan putar motor dapat dilihat melalui persamaan berikut :

$$\omega = \frac{V_t - I_a R_a}{K\phi} \quad (2.1)$$

Dimana :

ω = kecepatan putar motor (rad/sec)

V_t = tegangan masukan motor (V)

I_a = arus masukan motor (A)

R_a = hambatan jangkar motor (Ohm)

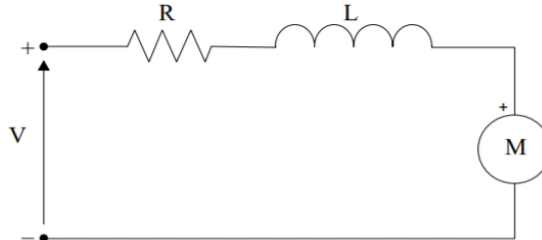
$K\phi$ = konstanta fluks magnetic

Torsi yang dihasilkan pada motor BLDC bergantung dari besarnya arus dan jumlah lilitan yang ada pada stator, kekuatan medan magnet yang ada pada magnet permanen di rotor, celah udara antara stator dengan rotor serta panjang dari magnet permanen pada rotor [14]. Besarnya nilai torsi yang dihasilkan dapat dilihat pada persamaan berikut :

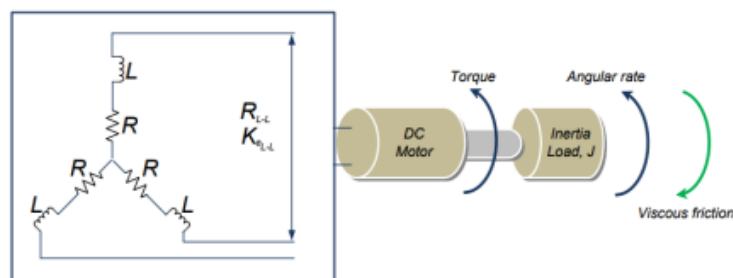
$$\tau = K\phi \cdot I_a \quad (2.2)$$

2.2.3 Pemodelan Matematika Motor BLDC

Berikut dijelaskan pemodelan matematika untuk Motor BLDC.



Gambar 2.2 Rangkaian motor listrik sederhana



Gambar 2.3 Diagram simetris BLDC

Dengan menggunakan hukum *kirchoff* maka :

$$V_s = Ri + L \frac{di}{dt} + e \quad (2.3)$$

Untuk kedudukan tetap (dimana kedudukan DC pada frekuensi 0) maka

$$V_s = Ri + e \quad (2.4)$$

Sedangkan, untuk kedudukan yang tidak stabil maka persamaan 2.3 diatur ulang dengan menambahkan BEMF, persamaan tersebut menjadi

$$e = -Ri - L \frac{di}{dt} + V_s \quad (2.5)$$

Dimana :

V_s = sumber tegangan DC

i = arus dinamo/arus motor

Berdasarkan hukum gerak *Newton* yang kedua, sifat matematis tergantung pada putaran pada sistem. Sistem pengaturan tersebut akan menghasilkan beban inersia (J)

$$J \frac{d\omega_m}{dt} = \sum T_i \quad (2.6)$$

$$T_e = k_f \omega_m + J \frac{d\omega_m}{dt} T_L \quad (2.7)$$

Dimana :

T_e = Putaran elektrik

k_f = gesekan konstan

J = rotor inersia

ω_m = kecepatan sudut

T_L = beban mekanik

Sehingga, putaran elektrik dan BEMF dapat dituliskan menjadi :

$$e = k_e \omega_m \quad (2.8)$$

Dan

$$T_e = k_t \omega_m \quad (2.9)$$

Dimana k_e adalah BEMF konstan dan k_t adalah putaran konstan. Sehingga, persamaan (2.5) dan (2.6) dapat dituliskan menjadi :

$$\frac{di}{dt} = -i \frac{R}{L} - \frac{k_e}{L} \omega_m + \frac{1}{L} V_s \quad (2.10)$$

$$\frac{d\omega_m}{dt} = i \frac{k_t}{J} - \frac{k_f}{J} \omega_m + \frac{1}{J} T_L \quad (2.11)$$

Persamaan (2.10) dan (2.11) tersebut dapat di evaluasi menggunakan transformasi *laplace* dengan membuat kondisi awal semuanya 0, maka untuk persamaan (2.10) :

$$L \left\{ \frac{di}{dt} = -i \frac{R}{L} - \frac{k_e}{L} \omega_m + \frac{1}{L} V_s \right\} \quad (2.12)$$

Berubah menjadi,

$$si = -i \frac{R}{L} - \frac{k_e}{L} \omega_m + \frac{1}{L} V_s \quad (2.13)$$

Dan untuk persamaan (2.11) :

$$L \left\{ \frac{d\omega_m}{dt} = i \frac{k_t}{J} - \frac{k_f}{J} \omega_m + \frac{1}{J} T_L \right\} \quad (2.14)$$

Berubah menjadi,

$$s\omega_m = -i \frac{k_t}{J} - \frac{k_f}{J} \omega_m + \frac{1}{J} T_L \quad (2.15)$$

Dengan $T_L = 0$ maka persamaan (2.15) diatas dapat ditulis menjadi

$$s\omega_m = -i \frac{k_t}{J} - \frac{k_f}{J} \omega_m \quad (2.16)$$

Dengan

$$i = \frac{s\omega_m + \frac{k_f}{J} \omega_m}{\frac{k_t}{J}} \quad (2.17)$$

Kemudian, persamaan (2.13) dapat ditulis menjadi

$$\left(\frac{s\omega_m + \frac{k_f}{J} \omega_m}{\frac{k_t}{J}} \right) \left(s + \frac{R}{L} \right) = -\frac{K_e}{L} \omega_m + \frac{1}{L} V_s \quad (2.18)$$

$$\left\{ \left(\frac{s^2 J}{k_t} + \frac{s k_f}{k_t} + \frac{s R J}{k_t} + \frac{k_f R}{k_t} \right) + \frac{K_e}{L} \right\} \omega_m = \frac{1}{L} V_s \quad (2.19)$$

$$V_s = \left\{ \frac{s^2 J + s k_f + s R J + k_f R + K_e k_t}{k_t} \right\} \omega_m \quad (2.20)$$

Sehingga, *transfer function* atau fungsi alih dapat dicari dengan menggunakan rasio kecepatan sudut ω_m dengan sumber tegangan V_s yaitu :

$$G(s) = \frac{\omega_m}{V_s} = \frac{k_t}{s^2 J + s k_f + s R J + k_f R + K_e k_t} \quad (2.21)$$

$$G(s) = \frac{\omega_m}{V_s} = \frac{k_t}{s^2 J + (R J + k_f L) + k_f R + K_e k_t} \quad (2.22)$$

Dengan mempertimbangkan asumsi berikut ini :

1. Gesekan konstan kecil, yaitu k_f mendekati 0, maka berate
2. $R J \gg k_f L$ dan
3. $k_e k_t \gg R k_f$

Dengan mengabaikan nilai 0, maka fungsi alih dapat disederhanakan menjadi :

$$G(s) = \frac{\omega_m}{V_s} = \frac{k_t}{s^2 J L + R J s + k_e k_t} \quad (2.23)$$

$$G(s) = \frac{\omega_m}{V_s} = \frac{\frac{k_t R}{k_e k_t R}}{\frac{s^2 J L R}{k_e k_t R} + \frac{R J s R}{k_e k_t R} + \frac{k_e k_t R}{k_e k_t R}} \quad (2.24)$$

$$G(s) = \frac{\frac{1}{k_e}}{\frac{R J}{k_e k_t R} s^2 + \frac{R J}{k_e k_t} s + 1} \quad (2.25)$$

Berdasarkan persamaan (2.25) diatas maka diperoleh rumus :

Untuk *mechanical (time constant)* :

$$\tau_m = \frac{R J}{k_e k_t} \quad (2.26)$$

Untuk *electrical (time constant)* :

$$\tau_e = \frac{L}{R} \quad (2.27)$$

Kemudian, substitusikan nilai τ_m dan τ_e ke persamaan (2.25), sehingga diperoleh

$$G(s) = \frac{\frac{1}{k_e}}{\tau_m \tau_e \cdot s^2 + \tau_m \cdot s + 1} \quad (2.28)$$

Pada pemodelan motor BLDC tidak jauh berbeda dengan motor DC, hal yang membedakannya yaitu terletak pada peletakan fasanya yang dimana akan mempengaruhi hasil keseluruhan pemodelannya. Pada persamaan (2.25) sampai (2.27) dapat dilihat perbedaan antara motor DC dengan motor BLDC. Perbedaan tersebut dapat dilihat melalui persamaan berikut :

Untuk *mechanical (time constant)* :

$$\tau_m = \sum \frac{RJ}{k_e k_t} = \frac{J \sum R}{k_e k_t} \quad (2.29)$$

Untuk *electrical (time constant)* :

$$\tau_e = \sum \frac{L}{R} = \frac{L}{\sum R} \quad (2.30)$$

Karena adanya susunan sistematis dan fase ke-3, persamaan untuk *mechanical time constant* dan *electrical time constant* menjadi :

$$\tau_m = \frac{J \cdot 3R}{k_e k_t} \quad (2.31)$$

$$\tau_e = \frac{L}{3R} \quad (2.32)$$

Dan untuk efek fase yaitu,

$$\tau_m = \frac{3 \cdot R \cdot \emptyset J}{\left(\frac{k_e(L-L)}{\sqrt{3}} \right) k_t} \quad (2.33)$$

$$\tau_m = \frac{3 \cdot R \cdot \emptyset J}{k_e \cdot k_t} \quad (2.34)$$

Dimana k_e merupakan nilai dari fase EMF tegangan konstan :

$$k_e = \frac{k_e(L-L)}{\sqrt{3}} \quad (2.35)$$

Terdapat hubungan antara k_e dan k_t yaitu menggunakan tenaga elektrik (sisi kiri) dan (sisi kanan) maka:

$$\sqrt{3} \times E \times I = \frac{2\pi}{60} \times N \times T \quad (2.36)$$

$$\frac{E}{N} = \frac{T}{I} \times \frac{2\pi \times 1}{60 \times \sqrt{3}} \quad (2.37)$$

$$k_e = k_t \times \frac{2\pi \times 1}{60 \times \sqrt{3}} \quad (2.38)$$

$$k_e = k_t \times 0.0605 \quad (2.39)$$

Dimana :

E = Percepatan (v/s)

N = Kecepatan putar motor (Rad/sec)

T = Medan magnet (Nm)

I = Arus listrik (A)

$$k_e = \left[\frac{V-secs}{rad} \right] : \text{Torsi elektrik}$$

$$k_e = \left[\frac{N-M}{A} \right] : \text{Torsi konstan}$$

Sehingga persamaan fungsi alih untuk motor BLDC dapat dtuliskan sebagai berikut dengan menghitung efek konstan dan fase :

$$G(s) = \frac{\frac{1}{k_e}}{\tau_m \cdot \tau_e \cdot s^2 + \tau_m \cdot s + 1} \quad (2.40)$$

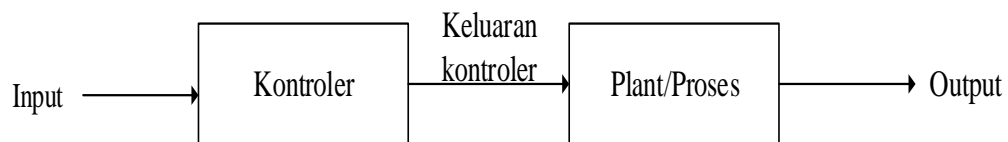
2.3 Sistem Kendali

Sistem adalah kumpulan dari beberapa komponen yang saling berhubungan dan menjadi satu kesatuan untuk menjalankan sebuah aksi. Kendali adalah mengawasi,

mengontrol, memerintahkan, mengendalikan. Jadi, sistem kendali adalah proses pengaturan atau pengendalian terhadap suatu besaran untuk mencapai harga atau *range* tertentu. Tujuan dari sistem kendali yaitu untuk mendapatkan titik optimal pada sebuah sistem yang dirancang. Sistem kendali dapat dibagi menjadi dua bagian yaitu sebagai berikut [13] :

2.3.1 Kalang Terbuka (*Open Loop*)

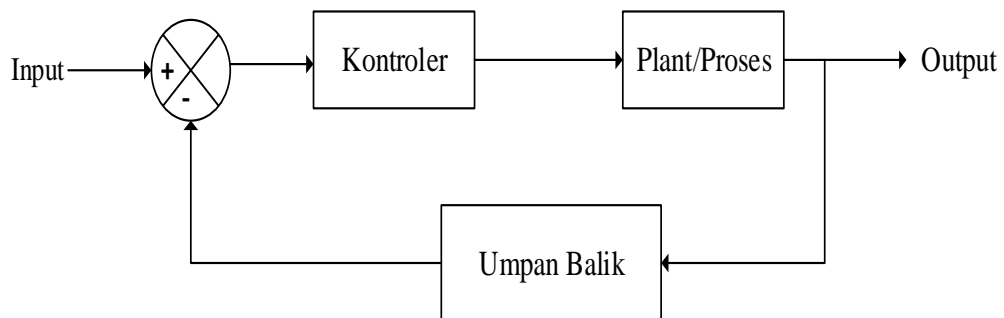
Sistem kendali kalang terbuka atau *open loop* adalah salah satu jenis sistem kendali yang aksi kontrolnya tidak tergantung dari keluaran sistem. Pada *open loop* saat *input* diberikan maka akan di proses lalu mengeluarkan *output* tanpa adanya *feedback* dari *output* dan apabila terjadi gangguan pada sistem maka tidak dapat dikoreksi, sehingga pada *open loop* masih membutuhkan peranan manusia sebagai operator.



Gambar 2.4 Sistem Kendali *Open Loop*

2.3.2 Kalang Tertutup (*Close Loop*)

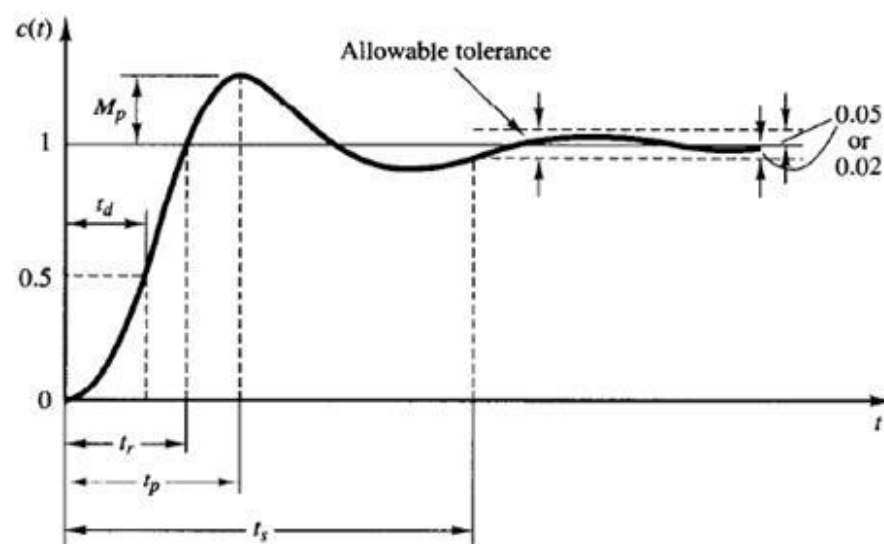
Sistem kendali kalang tertutup atau *close loop* adalah salah satu jenis sistem kendali yang aksi kontrolnya bergantung pada *output* sistem. Pada *close loop* saat *input* dimasukkan dan diproses maka akan ada *feedback* dari *output* nya, dimana jika terjadi kesalahan atau *error* maka akan diperbaiki agar nilai *output* sistem yang dihasilkan semakin mendekati nilai yang diharapkan.



Gambar 2.5 Sistem Kendali *Close Loop*

Perbedaan antara sistem kendali *open loop* dan *close loop* dapat dilihat pada Gambar 2.4 dan Gambar 2.5 yang menunjukkan ada atau tidaknya umpan balik *output* untuk dibandingkan kembali dengan *input*. Pada sistem *close loop* terdapat *summing point* dengan tanda (+/-) yang berfungsi untuk menjumlahkan sinyal yang masuk dan kemudian membandingkan besaran *set point* dengan sinyal hasil pengukuran [13].

2.3.3 Karakteristik Respon Sistem



Gambar 2.6 Kurva karakteristik respon sistem

Respon sistem merupakan salah satu parameter dari keberhasilan perancangan sistem yang dibuat, apabila respon sistem yang dihasilkan sesuai dengan keadaan yang diinginkan maka dapat dikatakan perancangan sistem tersebut berhasil. Perancangan suatu sistem pengendalian bertujuan untuk mendapatkan hasil kerja yang optimal dari sebuah sistem yang dikendalikan. Kurva karakteristik dari respon sistem dapat dilihat pada Gambar 2.6. Beberapa parameter yang dapat digunakan untuk mengetahui hasil kerja dari sistem pengendalian antara lain:

2.3.3.1 Waktu tunda / *delay time* (t_d)

Delay time adalah ukuran waktu yang dibutuhkan respon untuk mencapai setengah dari nilai akhir yang pertama.

2.3.3.2 Waktu naik / *rise time* (t_r)

Rise time adalah ukuran waktu yang menyatakan keadaan respon yang diukur dari respon 10% sampai 90% atau 0% sampai 100%.

2.3.3.3 Waktu puncak / *peak time* (t_p)

Peak time adalah ukuran waktu yang dibutuhkan respon untuk mencapai titik puncak *overshoot* yang pertama.

2.3.3.4 Waktu tunak / *settling time* (t_s)

Settling time adalah ukuran waktu yang dibutuhkan oleh kurva respon untuk mencapai keadaan tetap dalam *range* nilai akhir yang ditetapkan oleh persentase absolut berkisar 2% atau 5% dari respon *steady state*.

2.3.3.5 Overshoot Maksimal / *Maximum Overshoot* (M_p)

Maximum Overshoot adalah ukuran waktu yang menyatakan nilai puncak maksimum dari kurva respon yang merupakan perbandingan nilai maksimum respon yang melampaui nilai *steady state* dengan nilai *steady state* atau dapat dinyatakan dengan persamaan berikut :

$$M_p = \frac{\text{nilai puncak pertama} - \text{nilai akhir}}{\text{nilai akhir}} \times 100\%$$

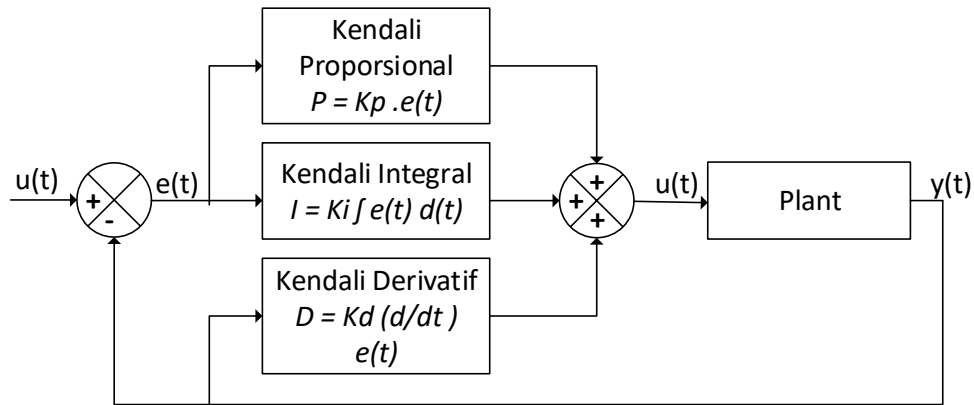
2.4 Sistem Kendali *Proportional Integral Derivative* (PID)

Sistem kendali PID yang tersusun dari tiga elemen pengendali yaitu *proportional*, *integral* dan *derivative*. Ketiga elemen tersebut dapat dikombinasikan menjadi beberapa tipe sistem kontroler sesuai dengan *plant* yang dibutuhkan. Elemen-elemen tersebut memiliki pengaruh yang berbeda-beda terhadap sistem yang di kontrol [13]. Elemen elemen tersebut yaitu K_p , K_d dan K_i , dibawah ini merupakan tabel yang menunjukkan pengaruh K_p , K_d dan K_i pada sistem *close loop* :

Tabel 2.1 Pengaruh respon pengendali PID

Respon Close loop	Rise Time	Overshoot	Settling Time	Steady State Error
Kp	Menurun	Meningkat	Perubahan kecil	Menurun
Ki	Menurun	Meningkat	Meningkat	Hilang
Kd	Perubahan kecil	Menurun	Menurun	Perubahan kecil

Dari Tabel 2.1 diatas dapat dilihat bahwa pengendali *proportional* (Kp) akan membuat waktu naik (*rise time*) menurun dan meningkatkan *overshoot* serta menurunkan *error* yang terjadi saat keadaan tunak. Pengendali *integral* (Ki) akan meningkatkan nilai *overshoot* dan *settling time*, menurunkan waktu naik atau *rise time* serta menghilangkan *error* yang terjadi di keadaan tunak.



Gambar 2.7 Diagram Blok Kendali PID

Dari blok diagram dapat diketahui bahwa penggabungan kendali proporsional, integral dan derivatif dapat dituliskan dalam bentuk *laplace* menjadi persamaan (2.41), dimana $u(t)$ merupakan hasil keluaran dari sistem kendali PID.

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_0^1 e(t) d(t) + \tau_d \frac{de(t)}{dt} \right) \quad (2.41)$$

Atau dalam bentuk *transfer function* dapat dituliskan sebagai berikut :

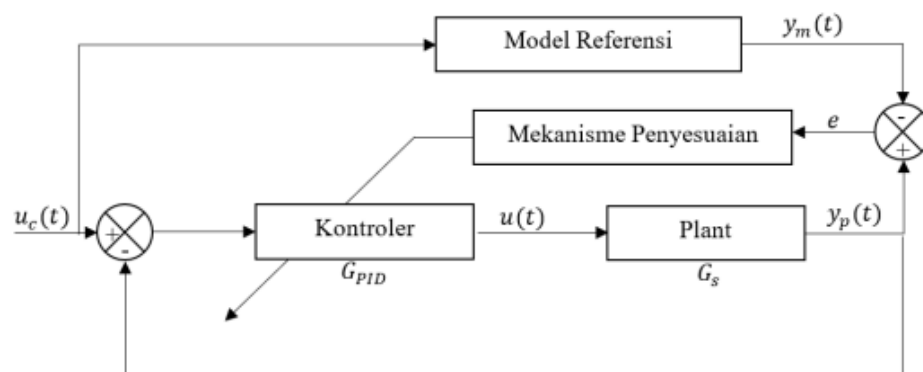
$$\frac{U(s)}{E(s)} = K_p \left\{ 1 + \frac{1}{\tau_i s} + \tau_d s \right\} = K_p + \frac{K_i}{s} + K_d s \quad (2.42)$$

Dimana K_p adalah penguatan *Proportional*, K_i adalah penguatan *integral* dan τ_i adalah waktu *integral*, K_d adalah penguatan *diferensial* dan τ_d adalah waktu waktu

derivative. K_p berfungsi untuk membuat kecepatan respon sistem meningkat, jika nilai K_p semakin besar maka respon dari sistem akan lebih cepat. K_i berfungsi untuk memperbaiki atau menghilangkan *error* yang terjadi pada *steady state*. K_d berfungsi untuk meningkatkan kestabilan sistem dan mengurangi *overshoot* [13][15].

2.5 Sistem Kendali *Model Reference Adaptive Controller* (MRAC)

Sistem kendali adaptif adalah sistem yang kendali yang dapat beradaptasi dengan kondisi sistem yang dikontrol baik itu dari luar sistem ataupun dari dalam sistem itu sendiri. Sistem kendali adaptif ini dapat memperbaiki kinerja sistem dan mempertahankan stabilitas sistem [13]. Sistem kendali adaptif sendiri dibagi menjadi beberapa tipe, salah satunya yaitu *Model Reference Adaptive Controller* (MRAC). MRAC merupakan salah satu skema adaptif dimana performansi keluaran sistem (proses) mengikuti dari performansi keluaran model referensinya. Parameter-parameter pengendali diatur melalui mekanisme pengaturan yang didasari pada *error* yang merupakan selisih antara keluaran *plant* dengan keluaran model referensi [8]. Untuk lebih jelas, dapat dilihat diagram blok dari MRAC pada Gambar 2.8.



Gambar 2.8 Diagram Blok Kendali MRAC

Dari diagram blok diatas dapat dilihat bahwa terdapat 2 *loop*, *loop* yang pertama yaitu *loop* umpan balik normal antara *output* proses dengan kontroler dan pada *loop* yang kedua adalah *loop* yang digunakan mekanisme pengaturan kontroler. Pada *loop* kedua ini dilakukan proses untuk *update* parameter – parameter kontroler

maupun parameter *plant* sesuai dengan skema adaptif yang digunakan. Pengaturan dilakukan dengan meminimalkan sinyal *error*, sehingga keluaran sistem (y) sesuai dengan keluaran model referensinya (y_m) [8].

2.5.1 Aturan *Massachusetts Institute of Technology* (MIT)

Aturan MIT adalah hukum penyesuaian parameter skalar yang diusulkan sekitar tahun 1960 yang digunakan untuk kendali adaptif. Aturan MIT merupakan turunan dari hasil perhitungan aturan adaptasi dengan menggunakan metode *gradient* [13]. Pada aturan MIT *loop* tertutup kontroler memiliki parameter yang dapat diatur berupa θ . Respon sistem *loop* tertutup ditentukan oleh model yang keluarannya dinotasikan dengan y_m dan output proses dinotasikan sebagai y . *Error* (e) merupakan selisih antara keluaran y dari sistem dengan keluaran dari model y_m . Pengaturan parameter dilakukan dengan meminimalkan fungsi kerugian (*The loss function* ($J(\theta)$)) [8] :

$$J(\theta) = \frac{1}{2} e^2 \quad (2.43)$$

Maka untuk mengubah parameter kendali θ dalam arah negatif terhadap gradien J .

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (2.44)$$

Koefisien γ adalah koefisien laju adaptasi. Dalam MRAC aturan MIT disebut juga dengan penguat adaptasi. Koefisien γ merupakan faktor pengali terhadap deret aritmatika. Dalam beberapa aplikasi ditemukan bahwa penentuan penguat adaptasi sangat penting dan nilainya tergantung pada tingkat kebutuhan sinyal [13].

Pada penelitian ini, penentuan model matematika MRAC dapat ditentukan menggunakan karakteristik orde dua dengan mengikuti persamaan berikut :

$$y_m(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.45)$$

Dimana

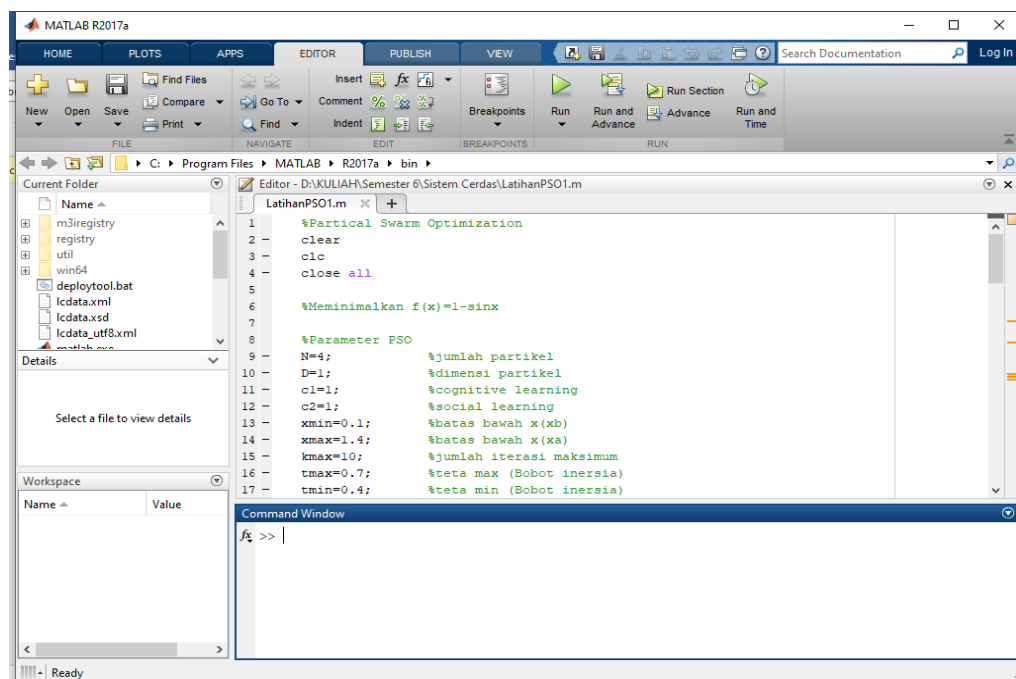
$$t_s = \frac{4}{\zeta\omega_n} \quad (2.46)$$

$$\omega_n = \frac{4}{\zeta t_s} \quad (2.47)$$

Besarnya nilai ζ dapat mempengaruhi respon model referensi. Nilai $\zeta < 1$ respon akan *underdamp*, $\zeta = 1$ respon akan *criticaldamp*, dan $\zeta > 1$ respon akan *overdamp*.

2.6 Matrix Laboratory (MATLAB)

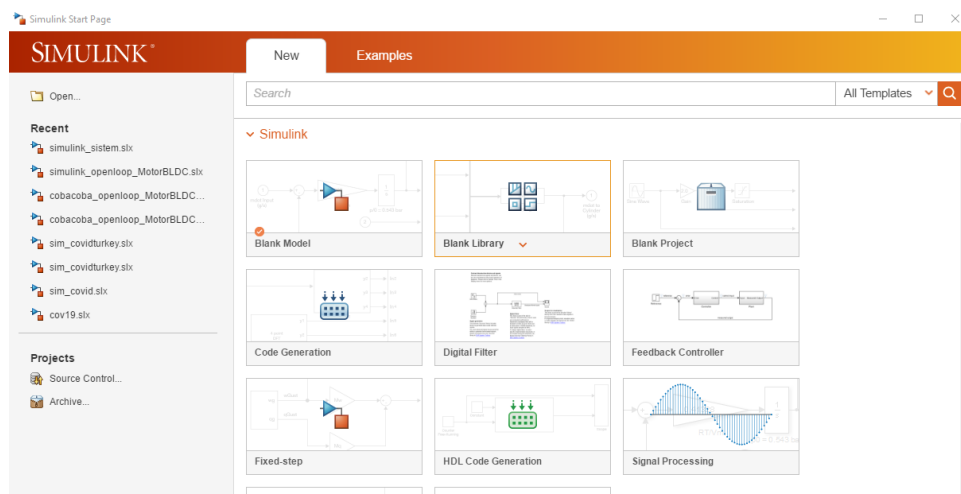
Matrix laboratory atau MATLAB merupakan sebuah *software* yang dapat melakukan suatu analisis dan juga komputasi. MATLAB dikembangkan oleh MathWorks. Pada MATLAB terdapat alat bantu (*toolbox*) yang menggunakan mesin simbolik MuPAD. *Toolbox* tersebut dapat memungkinkan akses terhadap kemampuan dalam komputasi aljabar.



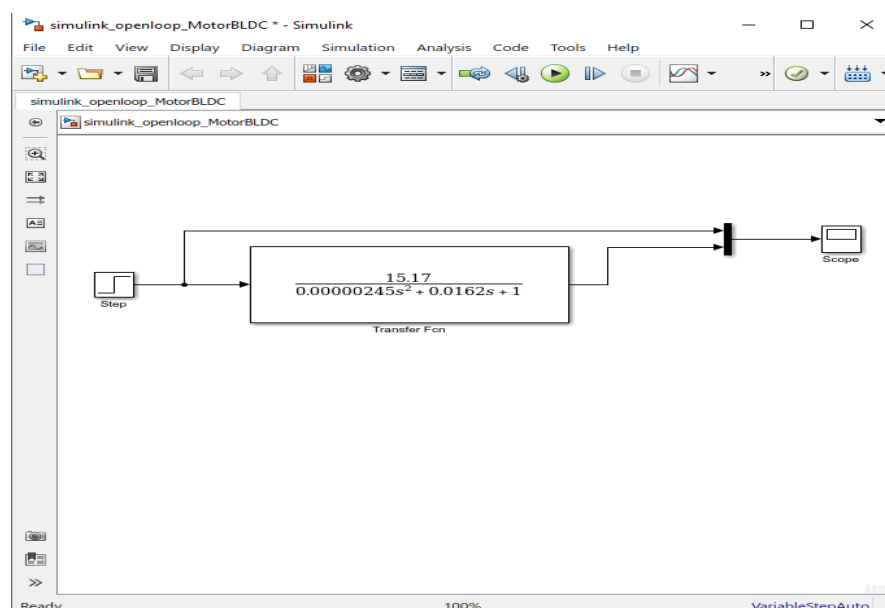
Gambar 2.9 Tampilan *software* MATLAB R2017a

Pada penelitian ini *toolbox* yang akan digunakan yaitu *Simulink* dan *PID tuner*. *Simulink* merupakan salah satu *toolbox* dari MATLAB yang berfungsi sebagai pemrograman grafis dan membuat simulasi sistem yang dinamik. *Simulink* dapat digunakan sebagai sarana pemodelan, simulasi dan juga analisis dari sebuah sistem

yang dynamin dengan antarmuka pengguna grafis. Analisis sistem yang dapat dilakukan oleh *Simulink* bisa berupa sistem linier dan juga sistem non-linier.

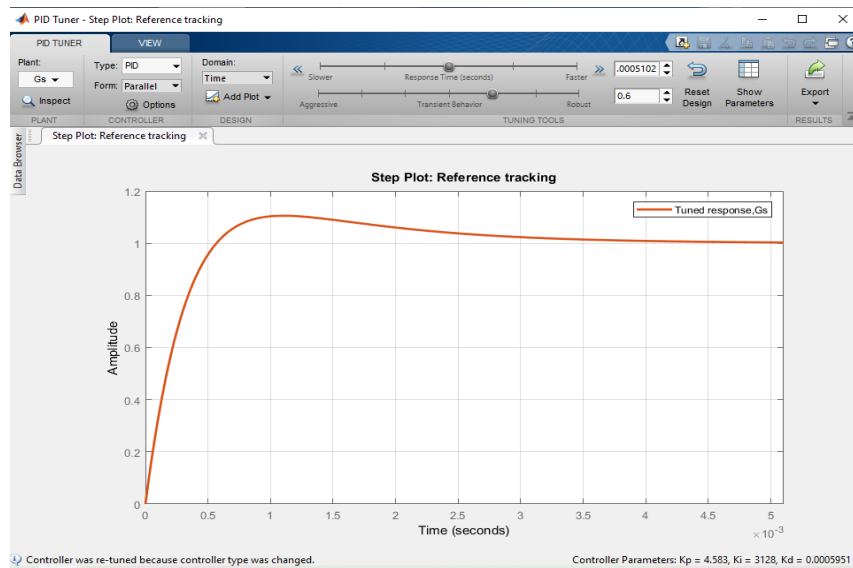


Gambar 2.10 Tampilan awal *toolbox Simulink*

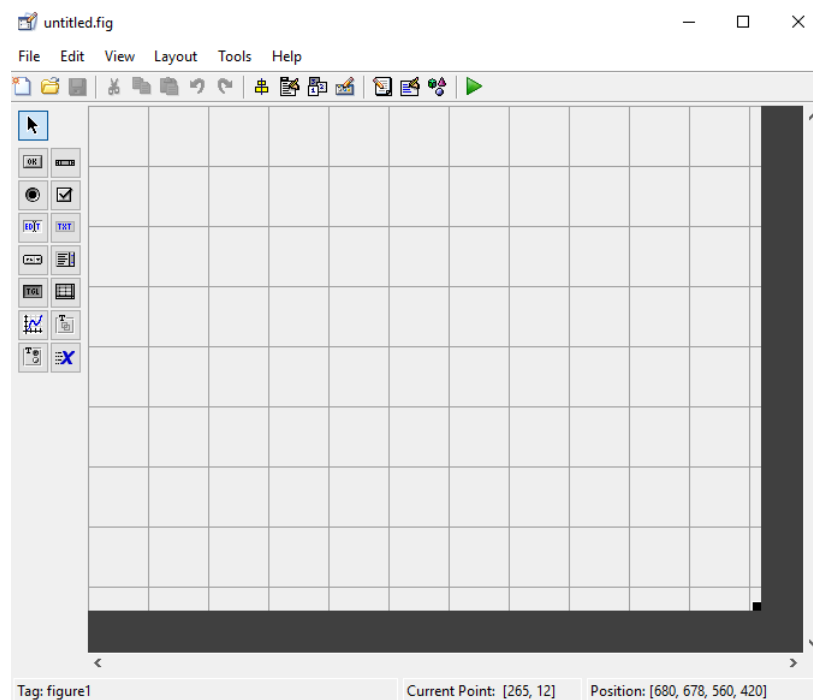


Gambar 2.11 Tampilan pemodelan simulasi sistem pada *Simulink*

PID *tuner* merupakan salah satu *toolbox controller* yang ada pada MATLAB yang memiliki fungsi sebagai tuning otomatis yang kemudian adapat dilihat hasil tanggapan dari sistem yang kendalikan. Pada PID *tuner* terdapat komponen yang dapat digunakan untuk mengatur kecepatan dan ketajaman dari tanggapan sistem sesuaiyang diinginkan yang kemudian dapat diketahui parameter-paramater PID yang dihasilkan.

Gambar 2.12 Tampilan PID *tuner*

GUI atau *Graphical User Interface* merupakan salah satu *toolbox* yang ada pada MATLAB yang berfungsi untuk menampilkan objek yang menyampaikan informasi dan mewakili tindakan. GUI dalam MATLAB dirangkum dalam sebuah aplikasi *guide* (*Graphical User Interface Builder*). Pada GUI terdapat komponen-komponen seperti tombol, kotak, teks, panel, slider, *pop-up menu*, grafik.



Gambar 2.13 Tampilan GUIDE

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat

Penelitian dilakukan di Laboratorium Pengukuran Besaran Listrik, Laboratorium Terpadu Jurusan Teknik Elektro Universitas Lampung. Penelitian ini dimulai dari bulan Januari 2023 sampai dengan bulan April 2023. Adapun pada Tabel 3.1 menunjukkan waktu kegiatan penelitian ini berlangsung.

Tabel 3.1 Waktu Penelitian

No.	Kegiatan	Januari				Februari				Maret				April			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Studi literatur																
2.	Penentuan konsep penelitian																
3.	Pengumpulan data																
4.	Perancangan simulasi sistem																
5.	Pengujian simulasi sistem																
6.	Pengambilan dan analisis data																
7.	Penulisan Laporan																

3.2 Alat dan Bahan

Adapun alat dan bahan yang digunakan dalam penelitian ini antara lain :

Tabel 3.2 Alat dan bahan

Nama alat dan Bahan	Justifikasi Penggunaan
Komputer, AMD Ryzen 3 3200U <i>with Radeon Vega Mobile Gfx</i> , RAM 8GB 64-bit	Sebagai tempat menyusun algoritma program serta menjalankan simulasi model sistem kendali <i>adaptive</i> PID pada motor BLDC.
<i>Software</i> MATLAB R2017a	Mensimulasikan model sistem kendali <i>adaptive</i> PID pada motor BLDC dari persamaan matematika dan juga melalui <i>toolbox Simulink</i> .

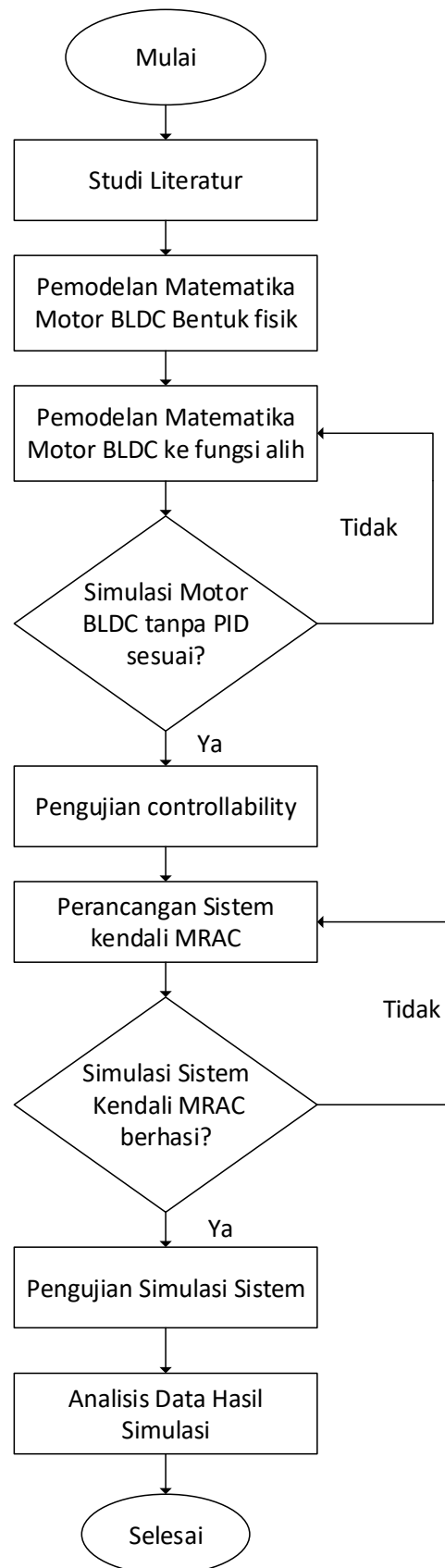
Pada penelitian ini, alat dan bahan yang digunakan terbagi menjadi 2 yaitu *hardware* yang berupa Laptop atau PC dan *software* yang berupa aplikasi MATLAB R2017a. MATLAB digunakan untuk mensimulasikan model sistem yang akan dirancang pada penelitian ini. Penggunaan alat dan bahan yang dipakai yaitu :

1. Listing program di *common windows* pada MATLAB digunakan untuk membuat program pemodelan matematika pada motor BLDC dan juga model referensi yang akan dibuat, serta digunakan untuk melakukan pengujian *controllability* pada sistem motor BLDC.
2. Penggunaan *toolbox Graphical User Interface* (GUI) digunakan untuk menampilkan hasil dari perancangan sistem kendali *adaptive* PID pada motor BLDC yang telah dirancang secara visual.
3. Penggunaan *toolbox Simulink* digunakan untuk merancang dan membuat model simulasi dari sistem motor BLDC sebelum ditambahkan *controller* dan juga sistem kendali *adaptive* PID pada motor BLDC. Pada *simulink* ini akan menampilkan bagaimana perancangan sistem simulasi kendali dan juga menjalankan simulasi tersebut yang kemudian akan ditampilkan hasilnya pada *scope*.
4. Penggunaan *toolbox PID tuner* digunakan untuk mensimulasikan sistem motor BLDC dengan kendali PID dengan cara *tuning* secara otomatis sehingga diperoleh nilai parameter-parameter K_p , K_i dan K_d yang diharapkan dan kemudian hasil tersebut akan dijadikan karakteristik respon yang diharapkan dalam perancangan sistem ini.

3.3 Diagram Alir Perancangan Simulasi Sistem Kendali

Tahapan dan metode yang digunakan pada penelitian ini dideskripsikan melalui diagram alir pada Gambar 3.1. Adapun Tahapan penelitian yang dilakukan adalah sebagai berikut :

1. Melakukan studi literatur mengenai teori-teori yang mendasari penelitian yang akan dilakukan.
2. Menentukan parameter dari motor BLDC yang akan digunakan dan memodelkannya ke dalam bentuk persamaan matematika dan juga fungsi alihnya.
3. Simulasi sistem *openloop* dari pemodelan motor BLDC yang telah dibuat dengan menggunakan *simulink* MATLAB.
4. Pengujian *controllability* pada sistem motor BLDC yang akan dikendalikan.
5. Menentukan karakteristik respon motor BLDC yang diinginkan menggunakan *PID tuner*.
6. Menentukan model referensi MRAC untuk motor BLDC menggunakan aturan MIT sehingga diperoleh persamaan fungsi alihnya untuk melakukan pembaruan parameter sinyal berupa nilai K_p , K_i dan K_d .
7. Perancangan simulasi sistem kendali MRAC pada *Simulink* MATLAB berdasarkan model motor BLDC yang akan dikendalikan dan model referensi MRAC yang telah ditentukan.
8. Simulasi sistem kendali MRAC yang telah dibuat dengan melakukan beberapa pemilihan gain adaptasi untuk mendapatkan karakteristik respon yang paling baik dan dapat mengikuti model referensi yang dirancang.
9. Pengambilan data dan analisa tanggapan sistem yang dihasilkan oleh simulasi sistem.
10. Penyusunan laporan tugas akhir berdasarkan data hasil dan analisis data.



Gambar 3.1 Diagram alir perancangan dan simulasi sistem kontrol

3.4 Parameter Motor BLDC

Pada penelitian ini motor BLDC yang digunakan adalah Motor BLDC dengan tipe *Maxon EC flat* Ø 45 mm, berikut adalah spesifikasi motor yang digunakan :

Tabel 3.2 Paramter Motor BLDC

Parameter Motor BLDC	Nilai	Satuan (dalam satuan mili)
Nilai Tegangan nominal		
<i>Nominal Voltage</i>	12.0	V
<i>No load Current</i>	150	A
<i>Nominal Speed</i>	2850	Rpm
<i>No load Speed</i>	4371	Rpm
<i>Nominal Torque</i>	59.1	Nm
<i>Nominal Current</i>	2.13	A
<i>Stall Torque</i>	0.255	Nm
<i>Starting Current</i>	10.0	A
<i>Maximum Efficiency</i>	77	%
Karakteristik		
<i>Terminal resistance phase to phase (R)</i>	1.1	Ω
<i>Terminal inductance phase to phase (L)</i>	0.50	H
Putaran konstan (k_t)	25.5	Nm/A
Kecepatan konstan (k_e)	35.4	Rpm/V
Gradiasi kecepatan/putaran (k_0)	17.6	Rpm/Nm
<i>Mechanical time constant (k_m)</i>	0.0162	S
<i>Rotor Inertia (J)</i>	82.5	gcm^2
<i>Number of phase</i>	3	

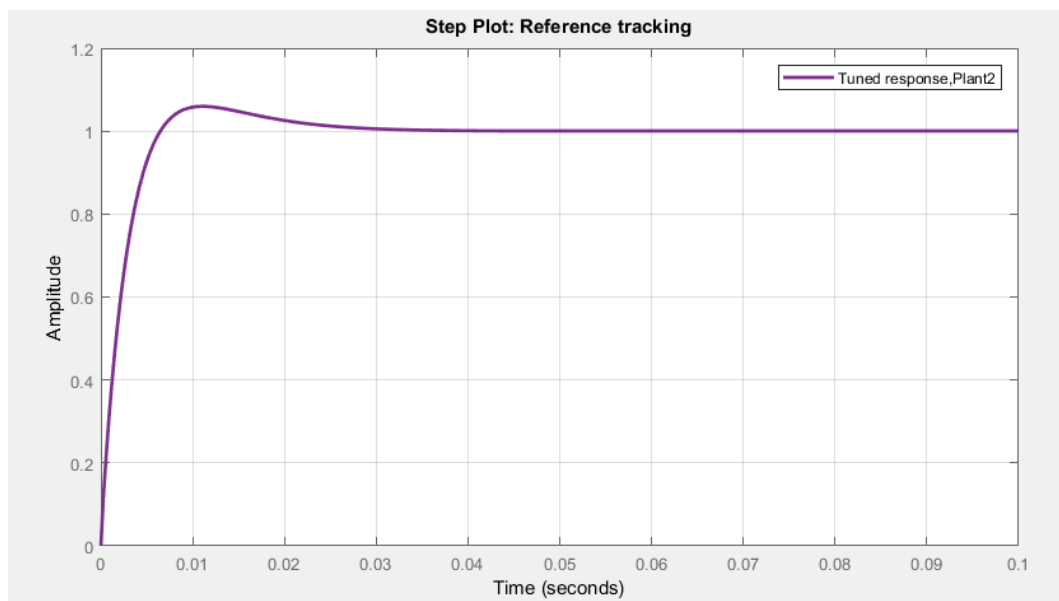
Pada penelitian ini, karakteristik utama dari Motor BLDC dengan tipe *Maxon EC flat* Ø 45 mm yang digunakan terdapat pada Tabel 3.2. Dari karakteristik tersebut maka perancangan simulasi sistem kendalu *adaptive* PID ini bertujuan untuk memberikan sinyal keluaran yang memiliki respon tanggapan yang sesuai dengan sifat yang diinginkan. Adapun respon tanggapan yang diinginkan pada penelitian ini yaitu memiliki *settling time* dan *rise time* yang singkat dan *overshoot* yang kecil. Sehingga performa dari motor BLDC ini dapat lebih efektif lagi.

3.4.1 Menentukan Pemodelan Matematika Motor BLDC

Parameter motor BLDC dapat dilihat pada Tabel 3.2 dan dari parameter-parameter tersebut maka dapat dibuat pemodelan matematika sistem kecepatan pada motor BLDC dalam bentuk fisik dan juga fungsi alihnya mengikuti persamaan (2.40). Data-data yang diperlukan untuk memodelkan sistem tersebut yaitu resistansi dan induktansi dari motor (R dan L), inersia dari rotor (J), dan konstanta waktu mekanis (k_m).

Sebelum menentukan model referensi dari sistem kendali MRAC-PID yang akan dibuat maka perlu dilakukan perhitungan dalam sistem kendali *closed loop* antara sistme kendali PID dengan *plant* ingin dikendalikan, agar karakteristik respon yang dihasilkan sesuai dengan karakteristik respon yang diinginkan.

Pada penelitian ini karakteristik respon yang diinginkan terlihat pada Gambar 3.4



Gambar 3.2 Respon Motor BLDC yang diharapkan

Dari Gambar 3.2 dapat diketahui bahwa karakteristik respon yang diharapkan dari hasil penelitian ini memiliki nilai *rise time* sebesar 0.00439 detik, *settling time* 0.0218 detik, *Overshoot* sebesar 5.97%.

3.4.2 Merancang Sistem Kendali MRAC-PID

Setelah didapatkan karakteristik respon yang diharapkan maka yang selanjutnya yaitu merancang sistem kendali MRAC-PID yang akan digunakan. Adapun alur perancangan kendali MRAC sebagai berikut :

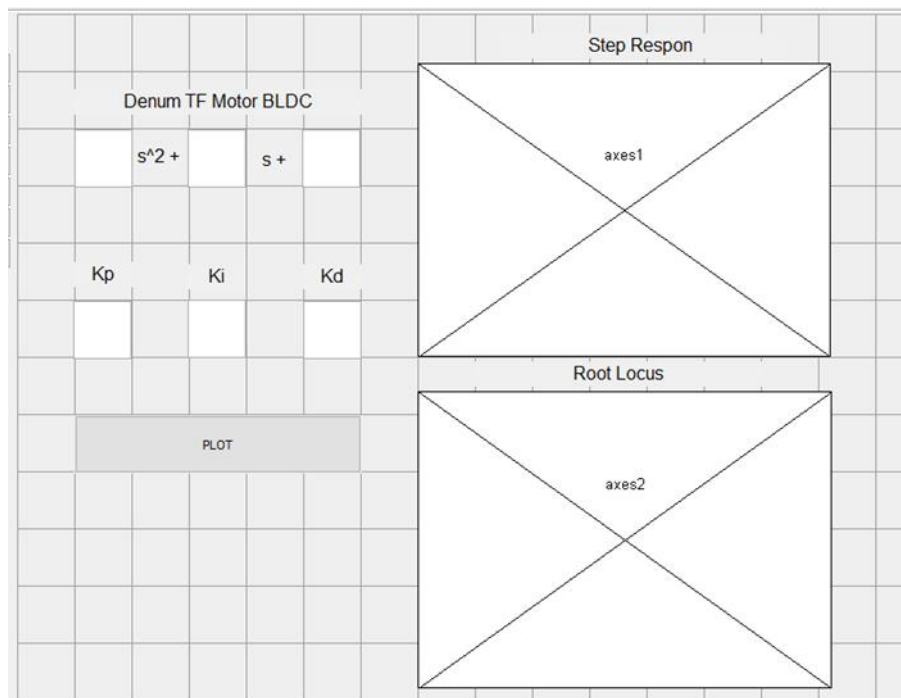
1. Membuat pemodelan matematika dari karakteristik motor BLDC dan memastikan bahwa sistem termasuk orde 1 ataupun orde 2.
2. Setelah membuat pemodelan matematika maka yang selanjutnya adalah membuat model referensi yang disesuaikan dengan parameter dari sistem kecepatan pada motor BLDC.
3. Langkah selanjutnya yaitu merancang pengendali MRAC menggunakan aturan MIT.
4. Setelah merancang pengendali MRAC, langkah selanjutnya yaitu menentukan nilai masing-masing *gamma* untuk mendapatkab respon yang diinginkan dengan metode *heuristik*. Rancangan metode ini diperoleh dengan cara perubahan parameter yang disesuaikan dengan kinerja dari *plant* yang akan dikendalikan.

3.5 Merancang *Graphical User Interface* (GUI)

Setelah melakukan perancangan sistem, maka sistem tersebut dapat dilihat secara visual menggunakan *toolbox* GUI. Adapun langkah-langkah dalam merancang GUI pada sistem *adaptive* PID yaitu :

1. Menentukan sistem *adaptive* yang telah dirancang menggunakan MRAC pada *Simulink* yang akan ditampilkan secara visual melalui GUI.
2. Menambahkan komponen-komponen yang ada pada GUI untuk membuat visualisasi sistem.
3. Memprogram sistem tersebut dalam GUI dan memasukkan nilai parameter-parameter (K_p, K_i, K_d) yang didapatkan dari hasil simulasi menggunakan *Simulink*.
4. Menjalankan program dari GUI tersebut sehingga dapat dilihat visualisasi dari sistem *adaptive* yang telah di simulasikan menggunakan *Simulink*.

Pada penelitian ini, GUI dari sistem yang akan ditampilkan dapat dilihat pada Gambar 3.3.



Gambar 3.3 Tampilan GUI Sistem Motor BLDC

DAFTAR PUSTAKA

- [1] R. Kandiban and R. Arulmozhiyal, "Speed Control of BLDC motor using Adaptive Fuzzy PID controller," *Procedia Engineering*, vol. 38, pp. 306-313, 2012.
- [2] M. D. A. and dkk, "Implementasi Dan Analisi Kendali Kecepatan Motor BLDC 1 kW Menggunakan Algoritma PID," *Jurnal ELTEK*, vol. 17, 2019.
- [3] D. Y. Agung and dkk, "Pengendalian Kecepatan Motor Brushless DC (BLDC) Menggunakan Metode Logika Fuzzy," *Jurnal Teknologi Techoscientia*, vol. 18, 2015.
- [4] M. A. Aziz, "Analisis Performa PID Pada Sistem Kontrol Kecepatan Motor BLDC," Surakarta, 2021.
- [5] M. Ali, "Pembelajaran Perancangan Sistem Kontrol PID dengan Software Matlab," *Jurnal Edukasi@elektro*, vol. 1, 2004.
- [6] T. Sudewo and dkk, "Desain dan Implementasi Kontrol PID Model Reference Adaptive Control untuk Automatic Safe Landing Pada Pesawat UAV Quadcopter," *Jurnal Teknik ITS*, vol. I, 2012.
- [7] H. P. Nurba and dkk, "Desain dan Implementasi Kendali PID Adaptif Pada Kecepatan Motor BLDC," vol. 2, 2015.
- [8] S. Amelia and dkk, "Perancangan dan Simulasi MRAC PID Control untuk Proses Pengendalian Temperatur Pada Continuous Stirred Tank Reactor (CSTR)," *Jurnal Teknik Pomits*, vol. III, pp. 23037-3539, 2014.

- [9] S. Erwin, "Kendali Adaptif Model Reference Adaptive Controller (MRAC) Pada Servo DC Menggunakan Estimator Recursive Least Square (RLS) dan Optimasi Genetik," *EECCiS*, 2008.
- [10] A. T. Wahyudi, "Desain dan Simulasi Kontrol Kecepatan Motor BLDC Menggunakan Metode Fuzzy - PID Controller Untuk Aplikasi Sepeda Motor Listrik," Institut Teknologi sepuluh November, Surabaya, 2018.
- [11] N. A. Fasalina, "Pengaturan Kecepatan Motor Brushless DC dengan Menggunakan Metode PID STR," Institut Teknologi Sepuluh November, Surabaya, 2018.
- [12] F. Dinansyar, "Pengaturan Kecepatan Motor Brushless DC Menggunakan Kontroler Fuzzy Berbasis Linier Quadratic Regulator," Institut Teknologi Sepuluh November, Surabaya, 2016.
- [13] A. Rosyid, "Perancangan dan Implementasi Sistem Kendali Adaptive PID Pada UAV Tipe Fixed Wing," Teknik Elektro Universitas Lampung, Bandar Lampung, 2020.