

PEMODELAN DAN SIMULASI *FUZZY LOGIC CONTROL* PADA MODEL *ARM ROBOT MANIPULATOR*

*M. Amirullah Akbar¹, Munadi²

¹Mahasiswa Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

²Dosen Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

Jl. Prof. Sudharto, SH., Tembalang-Semarang 50275, Telp. +62247460059

*E-mail: amir240891@yahoo.com

Abstrak

Strategi kontrol yang paling banyak digunakan dalam industri adalah proporsional integral derivatif (PID). Kepopuleran kontrol PID yaitu mampu memberikan kinerja yang kuat dalam berbagai kondisi operasi dan sebagian memiliki kesederhanaan dalam fungsi. Salah satu aplikasinya adalah untuk mengontrol posisi arm robot manipulator dengan menggunakan motor DC servo sebagai aktuator. Akan tetapi, kontrol PID memiliki kekurangan dalam hal mencapai keakuratan dan kepresisian yang tinggi pada waktu keadaan *transient* yang cepat. Oleh karena itu maka dibutuhkan suatu desain kontrol baru berupa *fuzzy logic control* (FLC) untuk mencapai target posisi yang diinginkan. Makalah ini menyajikan desain kontrol PID dan desain FLC. Desain kontrol PID dibuat dengan menggunakan otomatis *tuning* oleh *sisotool* dan desain FLC dibuat dengan *fuzzy logic toolbox*. Kedua desain kontrol tersebut selanjutnya diimplementasikan menggunakan Arduino mega 2560 dengan menggunakan potensiometer sebagai pengendali yang dihubungkan pada *Simulink Support Package* untuk *hardware* Arduino di MATLAB/Simulink. Selanjutnya, Sebuah desain kontrol yang lebih baik yaitu dengan menggunakan FLC diusulkan. Hasil simulasi ditunjukkan. Analisis kinerja menunjukkan efektivitas yang diusulkan *Fuzzy logic controller* dibandingkan dengan kontrol PID.

Kata kunci: *Fuzzy Logic Toolbox*, Kontrol *Fuzzy Logic*, Kontrol PID, Motor DC Servo, *Sisotool*

ABSTRACT

The most widely used control strategy in industry is proportional integral derivative (PID) control. The popularity of PID controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity. One of the application is to control arm robot manipulator by using Motor DC servo as actuator. However, PID control has shortcomings in terms of achieving high accuracy and precision during fast transient state. Therefore we need a new control design in the form of fuzzy logic control (FLC) to achieve the desired targets position. This paper presents the PID controller design and FLC design. PID controller design is created using the automatic tuning by *sisotool* and FLC design is made with *fuzzy logic toolbox*. Then, both of control design are implemented using the Arduino Mega 2560 as the controller using a potentiometer connected to the *Simulink Support Package* for Arduino hardware in MATLAB / Simulink. Furthermore, a better control design by using the proposed FLC. The simulation results are shown. Performance analysis shows the effectiveness of the proposed fuzzy logic controller compared to the PID controller.

Keyword: Motor DC servo, *Fuzzy Logic Controller*, *Fuzzy Logic Toolbox*, PID Controller, *Sisotool*

1. PENDAHULUAN

Sistem industri dengan efisiensi tinggi dan kinerja yang hebat telah mengambil keuntungan lebih dari teknologi robot. Sejumlah besar penelitian kontrol dan berbagai aplikasi kontrol disajikan selama tahun-tahun terakhir, berkonsentrasi pada pengendalian sistem robot. Bidang robot manipulator adalah salah satu aplikasi robot yang digunakan dalam industri, pendidikan dan kesehatan. Robot dapat bekerja di daerah berbahaya dan dalam keadaan tidak ramah lingkungan yang tidak dapat dicapai oleh manusia [1]. Misalnya, di reaktor nuklir atau suatu tempat dengan bahan kimia sangat berbahaya [1]. Oleh karena itu, pemodelan dan analisis sistem respon dari motor DC servo sebagai penggerak dalam robot manipulator dan menerapkan teknik kontrol seperti FLC guna menghasilkan gerakan yang halus, presisi dan akurat terhadap target sangat penting sebelum digunakan dalam situasi tersebut.

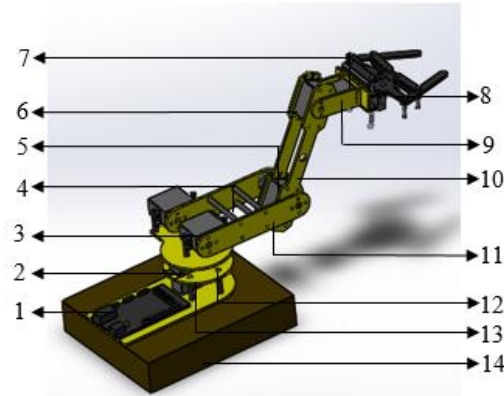
Makalah ini memiliki dua hal utama. Pertama, kontrol PID dirancang pada sistem orde tinggi menggunakan otomatis *tuning* PID dengan *sisotool*. Kedua, untuk sistem yang sama, FLC diusulkan dengan menggunakan *fuzzy logic toolbox* untuk merancang *fuzzy inference system* (FIS). Selanjutnya, kontrol PID didesain dan diimplementasikan dengan

maksud untuk mengetahui dan membuktikan kekurangan dari kontrol tersebut yang selanjutnya diusulkan sebuah kontrol yang lebih baik berupa FLC. Makalah ini terdiri dari beberapa bagian, pada bagian pertama berupa pendahuluan yang berisi tentang latar belakang dan tujuan dari penelitian. Bagian kedua menggambarkan desain dan parameter fungsi transfer motor DC servo. Bagian ketiga menjelaskan desain dan implementasi kontrol PID untuk motor servo DC. Bagian keempat menyajikan desain FIS untuk FLC pada motor DC servo dengan menggunakan *fuzzy logic toolbox* di Matlab/Simulink. Bagian kelima berupa kesimpulan [2].

2. DESAIN ARM ROBOT MANIPULATOR

2.1 Diagram CAD Model Arm Robot Manipulator

Dalam pembuatan desain CAD *arm robot manipulator* dilakukan pendesainan ulang terhadap *link*, *joint* serta *gripper*. Pembuatan desain robot tersebut dirancang dengan menggunakan software SolidWorks 2014. Pada Gambar 1. di bawah ini adalah skema hasil pembuatan desain *arm robot manipulator* dengan sudut pandang *isometric*.



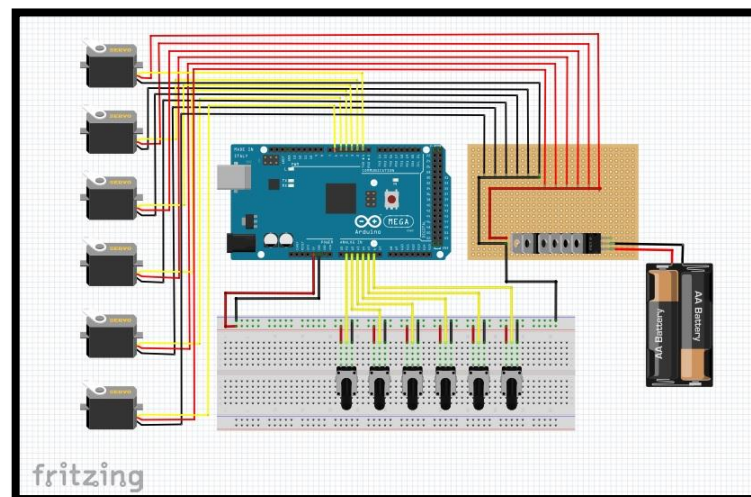
Gambar 1. Skema pembuatan simulator *arm robot manipulator*.

Keterangan Gambar 1 :

- | | |
|---------------------|------------------------------|
| 1 : Mikrokontrol | 8 : <i>Gripper</i> |
| 2 : <i>Joint 1</i> | 9 : <i>Link 3</i> |
| 3 : <i>Joint 2</i> | 10 : <i>Link 2</i> |
| 4 : Baut penghubung | 11 : <i>Link 1</i> |
| 5 : <i>Joint 3</i> | 12 : Penyangga <i>link 1</i> |
| 6 : <i>Joint 4</i> | 13 : <i>Bearing</i> |
| 7 : <i>Joint 5</i> | 14 : <i>Base Utama</i> |

2.2 Desain Hardware Elektronika Model Arm Robot Manipulator

Untuk sistem elektronika model *arm robot manipulator* ini terdiri atas beberapa komponen elektronika. Beberapa komponen elektronika tersebut disusun menggunakan *software* fritzing. Desain *wiring diagram* dapat dilihat pada Gambar 2.



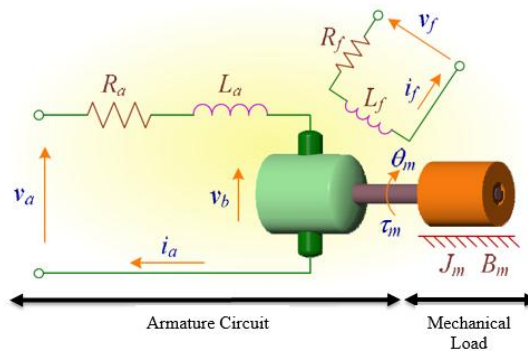
Gambar 2. Desain wiring diagram model arm robot manipulator.

2.3 Diskripsi Motor DC Servo

Motor DC servo digunakan sebagai aktuator *arm robot manipulator*. Gambar 3 mewakili model dari motor DC servo. Parameter dan nilai-nilai yang digunakan dalam simulasi dapat dilihat pada Tabel 1 [1].

Tabel 1. Parameter dan nilai motor DC servo.

Parameter	Value
Momen of inertia	$J_m = 0,000052 \text{ Kg.m}^2$
Friction coefficient	$B_m = 0,01 \text{ N.ms}$
Back EMF constant	$K_b = 0,235 \text{ V/rad s}^{-1}$
Torque constant	$K_a = 0,235 \text{ Nm/A}$
Electric resistance	$R_a = 2 \text{ ohm}$
Electric inductance	$L_a = 0,23 \text{ H}$
Input voltage	v_a
Back EMF voltage	v_b
Armature Current	i_a
Developed Torque	τ_m
Angle of motor shaft	θ
Load Torque	T_L

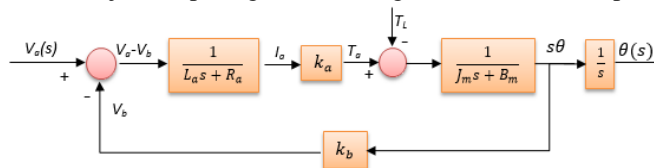


Gambar 3. Skema motor DC servo.

Tujuan dari pemodelan motor DC servo untuk mendekati keadaan motor servo yang sebenarnya [4]. Dengan memasukkan parameter, kita bisa mendapatkan fungsi transfer motor DC servo untuk melakukan kontrol posisi [1].

$$G_{posisi}(s) = \frac{\theta(s)}{V_a(s)} = \frac{19649}{s^3 + 201s^2 + 6290s} \tag{1}$$

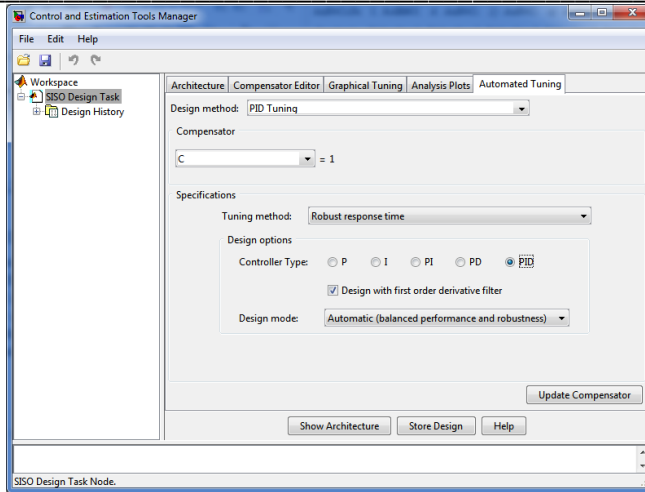
Diagram blok motor DC servo ditunjukkan pada gambar 4 dengan asumsi nilai nol pada torsi beban (T_L) [5].



Gambar 4. Diagram blok motor DC servo.

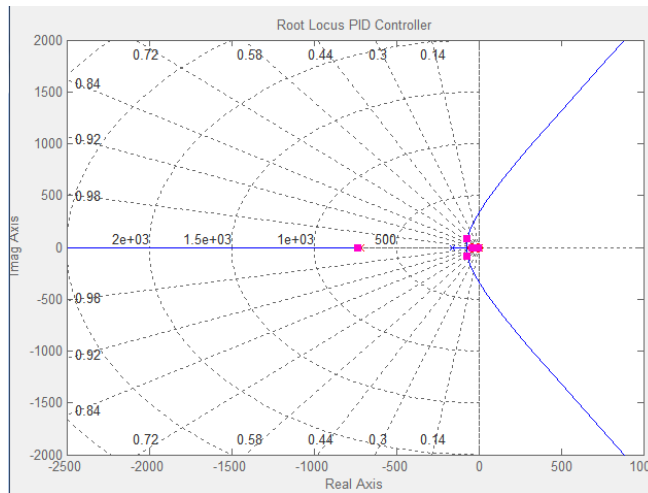
3. Desain Kontrol PID

Kontrol PID dirancang untuk sistem tatanan yang lebih tinggi dengan fungsi transfer dalam persamaan (1). Sebelum memulai simulasi dan implementasi kontrol PID, mendesain kontrol PID dengan menggunakan sisotool di MATLAB dan menganalisis stabilitas sistem respon dengan menggunakan metode root locus [4], [5], [6]. Tampilan antar muka sisotool pada MATLAB dengan menggunakan menu otomatis *tuning* PID seperti yang terlihat pada Gambar 5.



Gambar 5. Otomatis *tuning* PID menggunakan sisotool.

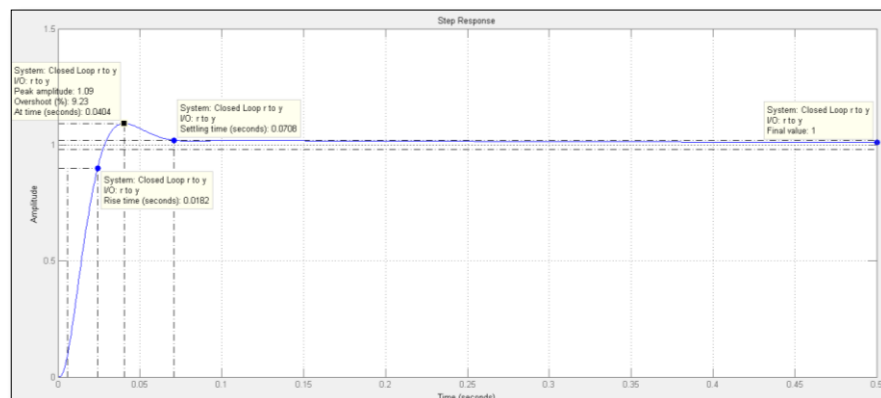
Dengan menggunakan otomatis PID tuning, kita bisa mendapatkan stabilitas root locus untuk sistem loop terbuka $G(s)$ ditunjukkan pada Gambar 6.



Gambar 6. Root locus kontrol PID.

Terdapat 5 daerah *pole* atau kutub di Gambar 6 yang mewakili stabilitas untuk sistem. Sebagian besar lokasi dari kutub terletak sangat dekat dengan sumbu imajiner sehingga sistem memiliki respon yang sangat sensitif. Terdapat dua daerah kutub yang tidak berada pada sumbu nyata dan sumbu imajiner. Kedua daerah *pole* tersebut menunjukkan bahwa terdapat *overshoot* pada sistem. Jika kutub melewati sumbu imajiner maka sistem menjadi tidak stabil. Secara keseluruhan berdasarkan grafik, sistem masih dalam kondisi stabil [5], [6].

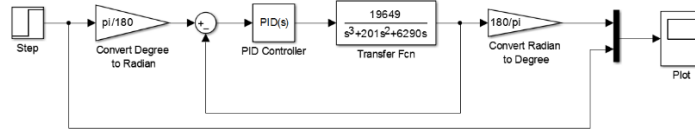
Step diberikan untuk desain kontrol PID. Gambar 7 menunjukkan respon sistem nya.



Gambar 7. *Step response* pada kontrol PID.

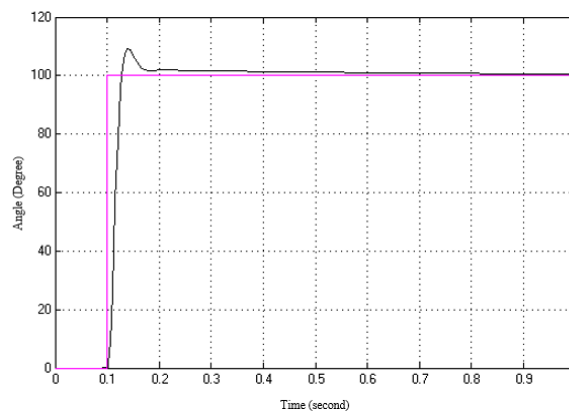
Berdasarkan Gambar 7, kita bisa mendapatkan parameter transien untuk sistem yaitu berupa waktu naik (*rise time*), persen *overshoot*, *settling time*, dan *steady state error*. Berdasarkan grafik pada gambar tersebut didapat nilai 0,0182 detik untuk *rise time* (T_r), 9,28% untuk *overshoot* (M_p), 0,0708 detik untuk *settling time* (T_s), dan 4% *steady state error* (E_{ss}).

Setelah mendesain PID, Gambar 8 menunjukkan model Simulink untuk simulasi kontrol PID. Selanjutnya, diberikan step posisi 100^0 dalam waktu 1 detik.



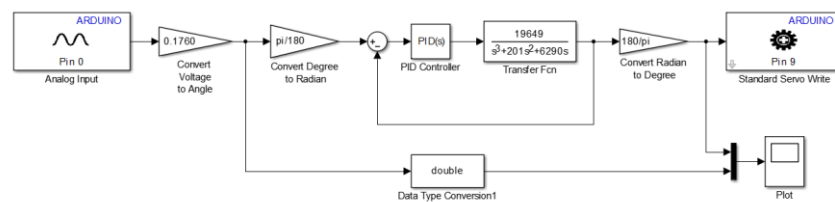
Gambar 8. Model Simulink untuk simulasi kontrol PID.

Berdasarkan model simulink pada Gambar 8 untuk simulasi kontrol PID maka dihasilkan Gambar 9 yang menunjukkan hasil simulasi.



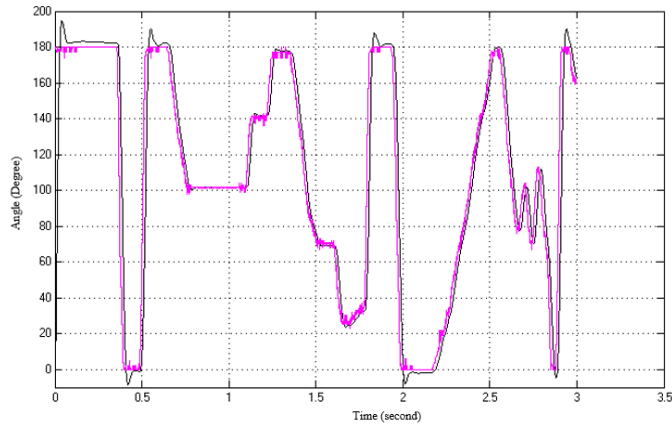
Gambar 9. Hasil simulasi kontrol PID.

Gambar 10. menunjukkan desain implementasi pada Simulink untuk kontrol PID dengan Arduino Mega 2560 via potensiometer dengan menggunakan *toolbox* Simulink berupa *Support Package for Arduino Hardware*.



Gambar 8. Model Simulink untuk implementasi kontrol PID.

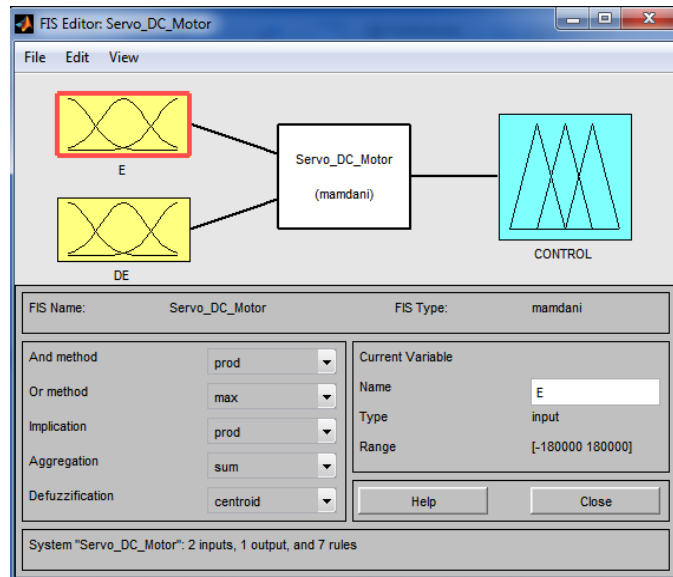
Berdasarkan model simulink untuk kontrol PID tersebut, pada Gambar 9 menunjukkan hasil implementasi.



Gambar 9. Hasil implementasi kontrol PID.

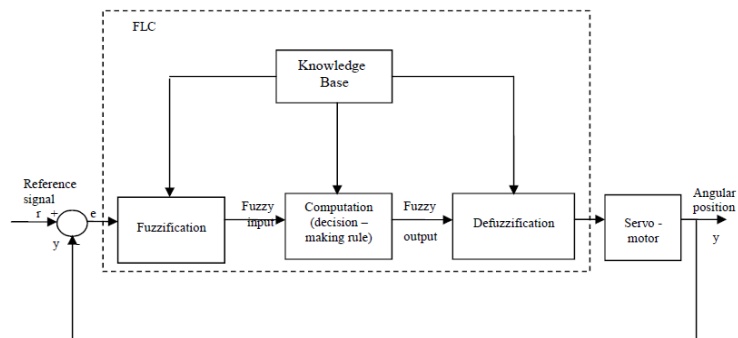
4. Fuzzy Logic Control (FLC)

Fuzzy Logic Control (FLC) atau Kontrol logika fuzzy dirancang untuk sistem tatanan yang lebih tinggi dengan fungsi transfer dalam persamaan (1). Sebelum memulai simulasi, kita mendesain FIS dengan menggunakan *fuzzy logic toolbox* di Gambar 10 kemudian disimulasikan di Simulink untuk menganalisis stabilitas sistem respon dengan membaca secara manual berdasarkan pada grafik tersebut.



Gambar 10. FIS editor.

Struktur FLC dasar seperti yang ditunjukkan pada Gambar11 berisi empat langkah yaitu fuzzifikasi, basis pengetahuan, pengambilan keputusan dan unit defuzzifikasi [7], [8].

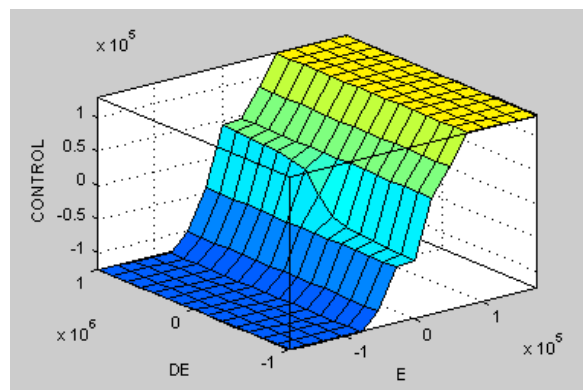


Gambar 11. Struktur dasar FLC [8].

Dalam desain simulasi ini, bertujuan untuk mengontrol posisi motor. Basis pengetahuan berisi seperangkat aturan yang membangun logika pengambilan keputusan [8]. Terdapat 7 aturan yang digunakan pada kontrol yang didasarkan pada pengalaman manusia dan informasi adalah:

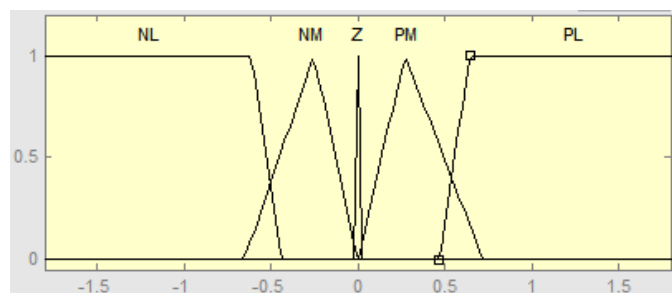
1. *If E is PL then CONTROL is PL*
2. *If E is NL then CONTROL is NL*
3. *If E is Z and DE is N then CONTROL is NM*
4. *If E is Z and DE is P then CONTROL is PM*
5. *If E is Z then CONTROL is Z*
6. *If E is NM then CONTROL is NM*
7. *If E is PM then CONTROL is PM*

Setelah merancang aturan tersebut, kemudian didapatkan *surface viewer* seperti yang terlihat pada Gambar 12 yang mewakili aturan FLC.

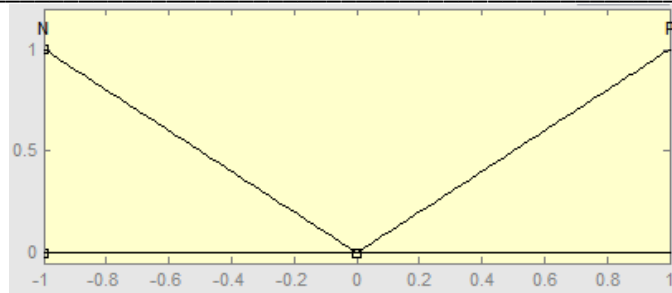


Gambar 12. 3D FLC surface.

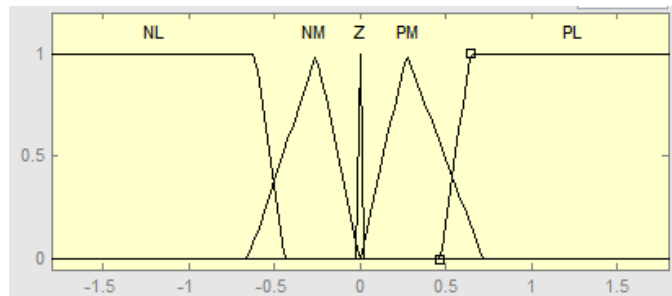
Terdapat dua *input* dan satu *output* di FIS. Dua *input* berisi error (E) dan delta error (DE) sementara *output* berisi CONTROL berupa sinyal kontrol yang masuk ke dalam fungsi transfer [2]. Fungsi keanggotaan atau *membership function* FLC untuk kedua *input* tersebut ditunjukkan pada Gambar 13 dan Gambar 14, dan fungsi keanggotaan *output* ditunjukkan pada Gambar 15. Untuk fungsi keanggotaan error (E) sebagai masukan dan fungsi keanggotaan CONTROL sebagai *output* terdiri dari variabel linguistik berupa NL berarti *Negative Large*, NM berarti *Negative Medium*, Z berarti *Zero*, PM berarti *Positive Medium*, dan PL berarti *Positive Large*. Untuk delta error (DE) sebagai *input*, memiliki variabel linguistik N berarti *Negative* dan P berarti *Positive*[7], [8].



Gambar 13. Membership function error (E) sebagai input.

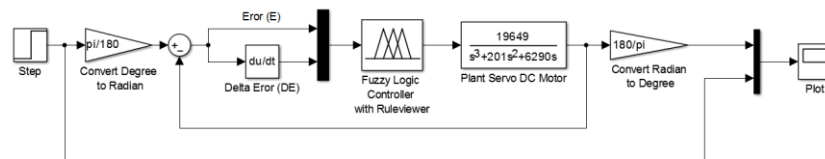


Gambar 14. Membeship function delta error (DE) sebagai input.



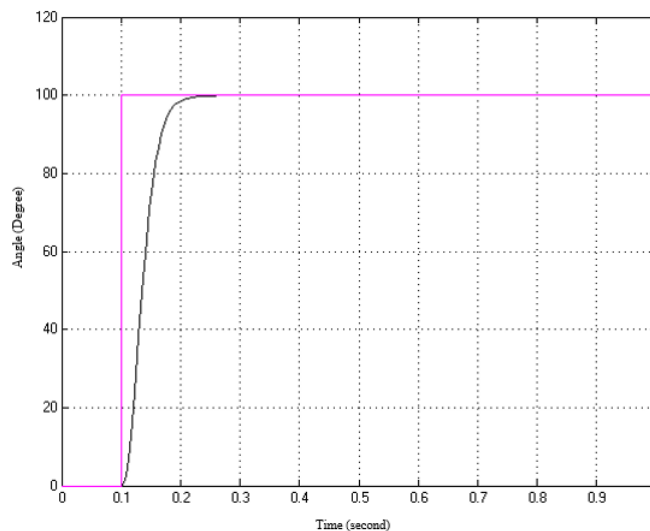
Gambar 15. Membeship function CONTROL sebagai output.

Setelah mendesain dan membangun FIS, Gambar 16 menunjukkan model Simulink untuk simulasi FLC. Selanjutnya, diberikan step posisi 100^0 dalam waktu 1 detik.



Gambar 16. Model simulink untuk simulasi FLC.

Berdasarkan model Simulink untuk simulasi FLC, pada Gambar 17 menunjukkan hasil simulasi.



Gambar 17. Model simulink untuk simulasi FLC.

Berdasarkan Gambar 17, kita bisa mendapatkan parameter transien untuk sistem berupa *rise time*, persen *overshoot*, *settling time*, dan *steady state error*. Untuk sistem, didapatkan nilai sebesar 0,07 detik untuk *rise time* (T_r), 0% untuk *overshoot* (M_p), 0,12 detik untuk *settling time* (T_s), dan 0% *steady state error* (E_{ss}).

Tabel 2 berikut menyajikan perbandingan parameter sistem respon yang meliputi *rise time* (T_r), persen *overshoot* (M_p), *settling time* (T_s) dan *steady state error* (E_{ss}) untuk kontrol PID dan *fuzzy logic controller* (FLC).

Tabel 2. Parameter dan nilai motor DC servo.

<i>Controller Type</i>	T_r (Sec)	M_p (%)	T_s (Sec)	E_{ss} (%)
<i>PID Controller</i>	0,0182	9,23	0,0708	4
FLC	0,07	0	0,12	0

5. Kesimpulan

Desain sistem motor DC servo menggunakan kedua kontrol PID dan FLC telah disajikan . Desain kontrol PID telah berhasil dilaksanakan tetapi masih terdapat overshoot. Jadi, desain kontrol yang lebih baik harus diusulkan untuk mengendalikan gerakan yang halus pada gerakan model lengan robot manipulator. Kinerja kontrol yang lebih baik, *robustness* dan stabilitas keseluruhan dapat dipenuhi dari kontrol berupa FLC. FLC memiliki stabilitas sistem yang lebih baik, overshoot kecil, dan respon yang cepat [7], [9].

Meskipun, kontrol PID menghasilkan respon dengan *rise time* yang lebih rendah dibandingkan dengan FLC, tetapi menghasilkan *error steady state* yang tinggi karena sistem berosilasi pada periode transien .Sistem dengan kontrol PID tersebut memiliki osilasi dengan overshoot yang sangat tinggi yaitu sebesar 9,23 % sehingga mampu menyebabkan kerusakan pada kinerja sistem. FLC yang diusulkan dapat secara efektif menghilangkan osilasi berbahaya dan memberikan kelancaran dalam periode transient. Oleh karena itu, dapat disimpulkan bahwa kontrol PID tidak dapat digunakan untuk gerakan presisi kontrol model lengan robot manipulator karena berosilasi. Hasil menunjukkan bahwa FLC berbasis desain yang diusulkan dapat menjadi pilihan yang lebih baik untuk implementasi berikutnya.

6. Referensi

- [1] M. A. Rashidifar, A. A. Rashidifar, D. Ahmadi, "Modeling and Control of 5DOF Robot Arm Using Fuzzy Logic Supervisory Control," *International Journal of Robotics and Automation (IJRA)*, Vol. 2, No. 2, June 2013, pp. 56~68.
- [2] S.R.Vaishnav, Z.J.Khan, "Design and Performance of PID and Fuzzy Logic Controller with Smaller Rule Set for Higher Order System," *Proceedings of the World Congress on Engineering and Computer Science, WCECS 2007*, October 24-26, 2007, San Francisco, USA
- [3] C. Urrea and J Kern, "A New Model for Analog Servo Motors", *Canadian Journal on Automation, Control and Intelligent Systems*, Vol. 2, No. 2, March 2011.
- [4] R. S . Burns, "Advances Control Engineering", Jordan Hill, 2001.
- [5] C. L. Philips and R. D. Harbor, "Feedback Control Systems", Prentice-Hall, 2000.
- [6] K. Ogata, "Modern Control Engineering", Prentice-Hall, 1970.
- [7] S. N. Sivanandam, S. Sumathi and S. N. Deepa, "Introduction to Fuzzy Logic using MATLAB", Springer, 2007.
- [8] S. F. Abd-Alkarim "Application of Fuzzy Logic in Servo Motor", *Al-Khwarizmi Engineering Journal* 2007, Vol.3, No.2, pp8 -16.
- [9] Gaurav, Amrit Kaur, "Comparison between Conventional PID and Fuzzy Logic Controller for Liquid Flow Control: Performance Evaluation of Fuzzy Logic and PID Controller by Using MATLAB/Simulink," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, ISSN: 2278-3075, Volume-1, Issue-1, June 2012.