

KONVERSI FUZZY LOGIC CONTROLLER KE NEURAL NETWORK CONTROLLER UNTUK KENDALI POSISI SUDUT MOTOR DC

Fuzzy Logic Controller Conversion to Neural Network Controller for DC Motor Angle Position Control

Ananta Triantama Putra¹, Agung Surya Wibowo, S.T., M.T.², Ir. Porman Pangaribuan, M.T.³

^{1,2,3}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom
¹nordeforce@student.telkomuniversity.ac.id, ²agungsw@telkomuniversity.ac.id,
³porman@telkomuniversity.com

Abstrak

Kendali posisi merupakan salah satu dari sistem kendali yang digunakan untuk mengendalikan posisi dari suatu objek, dalam hal ini yang dikendalikan adalah posisi sudut motor DC. Penelitian ini dilakukan untuk merancang alat yang dapat mengontrol sudut motor DC. Alat ini dirancang menggunakan *microcontroller* Arduino UNO. Proses pengendalian dari sudut motor DC dilakukan dengan menggunakan metode kendali *fuzzy logic controller* (FLC) dan *neural network controller* (NNC). Hasil yang diperoleh dari metode kendali FLC adalah *error*, Δ *error*, dan *output*. Hasil tersebut kemudian akan digunakan sebagai *training set* untuk pelatihan sistem dengan menggunakan metode kendali NNC dengan algoritma *back propagation*. Respons dari sistem motor DC akan menghasilkan *error steady state* < 2% dengan menggunakan sistem kendali FLC, serta alat ini diharapkan akan memiliki respons yang sama dengan respons sistem jika menggunakan sistem kendali NNC.

Kata Kunci : *Kendali Posisi, Fuzzy Logic Controller, Neural Network Controller, Back Propagation*

Abstract

Position control is one of the control systems used to control the position of an object, in this case what is controlled is the angular position of the DC motor. This research was conducted to design a device that can control the angle of a DC motor. This tool is designed to use an Arduino UNO microcontroller. The control process from the angle of the DC motor is carried out using 2 control method, the fuzzy logic controller (FLC) and the neural network controller (NNC). The results obtained from the FLC control method are error, Δ error, and output. The results will then be used as a training set for system training using the NNC control method with back propagation algorithm. The response from the DC motor system will produce a steady state error of <2% using the FLC control system, and this device is expected to have the same response as the system response when using the NNC control system.

Keywords: *Position Control, Fuzzy Logic Controller, Neural Network Controller, Back Propagation*

1. Pendahuluan

Kendali posisi merupakan salah satu dari sistem kendali yang digunakan untuk mengendalikan posisi dari suatu objek, dalam hal ini motor DC. Dalam proses pengendalian ini diperlukan sistem yang stabil. Sistem yang stabil merupakan sebuah sistem yang mempunyai respons yang cepat dan memiliki performa yang bagus. Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi mekanik. Untuk mengendalikan posisi sudut motor DC diperlukan metode kendali. Metode kendali tersebut adalah *PID Controller*, *Fuzzy Logic Controller* (FLC) dan *Neural Network Controller* (NNC).

Kontrol posisi sudut motor DC umumnya menggunakan metode *PID Controller* atau menggunakan metode FLC. Namun belum ada yang menggunakan NNC dalam mengontrol posisi sudut motor DC. Proses pengendalian dari sudut motor DC dilakukan dengan menggunakan metode kendali *fuzzy logic controller* (FLC) dan *neural network controller* (NNC). Hasil yang diperoleh dari metode kendali FLC adalah *error*, Δ *error*, dan *output*. Hasil tersebut kemudian akan digunakan sebagai *training set* untuk pelatihan sistem dengan menggunakan metode kendali NNC dengan algoritma *back propagation*.

2. Dasar Teori

2.1 Motor DC

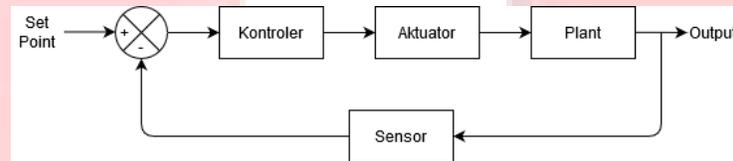
Motor DC merupakan salah satu jenis aktuator berupa mesin listrik yang mengubah energi listrik menjadi energi mekanik berupa gerakan motor. Kelebihan dari motor DC adalah torsi yang dihasilkan tinggi atau percepatan tetap untuk kisaran kecepatan yang luas, kecepatan dapat dikontrol dengan mudah[7].

2.2 Sistem Kontrol

Sistem kontrol adalah sebuah proses pengaturan atau pengendalian terhadap satu atau beberapa variabel atau parameter sehingga berada pada suatu *range* tertentu[2]. Secara umum sistem kontrol terbagi atas dua bagian, yaitu

sistem kontrol lup terbuka (*open loop control system*) dan sistem kontrol lup tertutup (*closed loop control system*). Sistem kontrol lup terbuka adalah sistem kontrol yang keluaran sistem tidak berpengaruh pada aksi pengendalian sistem. Pada sistem ini, keluaran tidak diumpan balikkan untuk dibandingkan dengan masukan sistem[5].

Sedangkan sistem kontrol lup tertutup adalah sistem kontrol yang keluaran sistem berpengaruh pada aksi pengendalian sistem. Pada sistem ini, keluaran diumpan balikkan untuk dibandingkan dengan masukan sistem[5]. Diagram blok sistem akan ditunjukkan pada Gambar 2 – 1.



Gambar 2 - 1. Diagram Blok Sistem Lup Tertutup

2.3 Fuzzy Logic Controller

Fuzzy logic merupakan sebuah logika yang memiliki nilai samar antara benar atau salah. Dalam teori *fuzzy logic*, sebuah nilai bisa bernilai salah dan bisa bernilai benar secara bersamaan. Namun besar nilai kebenaran dan besar nilai kesalahan tergantung pada bobot keanggotaan yang dimilikinya.

2.3.1. Dasar Kontrol Fuzzy

Dalam proses pengontrolan *fuzzy* terdiri atas tiga komponen, yaitu fuzzifikasi (*fuzzification*), evaluasi aturan (*inference*), dan defuzzifikasi (*defuzzification*). Fuzzifikasi diperlukan untuk mengubah masukan *crisp* yang bersifat bukan *fuzzy* ke dalam himpunan *fuzzy*. Data yang terbentuk *crisp*, dipetakan menjadi nilai linguistik pada semesta pembicaraan tertentu yang selanjutnya dinamakan masukan fuzzy[3].

Evaluasi aturan (*inference*) sebagai acuan untuk menjelaskan gabungan antara variabel-variabel masukan dan keluaran yang mana variabel yang diproses dan yang dihasilkan berbentuk *fuzzy*. *Input* dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*. Sedangkan *output* yang dihasilkan merupakan suatu bilangan domain himpunan *fuzzy* tersebut.

2.3.2. Desain Fuzzy Logic Controller

Teknologi *Fuzzy Logic Controller* (FLC) telah banyak yang menggunakannya dan mampu menggunakannya secara efektif dalam aplikasi manufaktur. FLC adalah sebuah metode untuk membuat sebuah teknologi menjadi lebih canggih. Model FLC merupakan model yang terbentuk dari logika matematika “*if-then*”. FLC memiliki desain dua variabel *input* yaitu variabel *error* dan variabel $\Delta error$ dan memiliki satu variabel *output*. FLC memiliki respons sistem yang dinamis berdasarkan dengan *input* referensi sistem[1].

2.4 Neural Network Controller

Artificial neural network (ANN) adalah sebuah jaringan saraf tiruan yang diadopsi dari saraf biologis manusia. Jaringan saraf tiruan pertama kali pada tahun 1943 oleh McCulloch dan Walter Pitts, mereka memiliki prinsip jaringan saraf tiruan dapat menghitung fungsi algoritma dan logika[4].

Untuk menggunakan NNC diperlukan data-data yang akan digunakan untuk melatih NNC. Pada saat proses *training Neural Network Controller*, data-data yang didapatkan dari sistem yang tidak menggunakan kontroler maupun sistem yang menggunakan kontroler akan dimasukkan ke *Neural Network Controller*. Data-data tersebut disebut sebagai bobot pada masing-masing *neuron* pada *input layer*. Bobot-bobot tersebut akan diperbarui secara otomatis dengan menggunakan metode *back propagation*.

2.5 Algoritma Back Propagation

Algoritma pelatihan *back propagation* merupakan sebuah metode *supervised learning algorithm* untuk umpan maju *multilayer neural network*. Data eror pada *output layer* dihitung menggunakan *ouput* dan *target output*. Lalu eror akan dipropagasi kembali ke *intermediate layer*, memungkinkan eror masuk ke *layer* ini dan memperbaiki bobot. Pada dasarnya, proses *error back propagation* terdiri dua lapisan *network*, yaitu *forward pass* dan *backward pass*. Pada *forward pass*, vektor *input* dimasukkan ke dalam *network* dan efeknya menyebar dari *layer* ke *layer*. Selama proses *forward pass*, nilai bobot disinapsis semuanya tetap. Sebaliknya, selama proses *backward pass* nilai bobot disinapsis disesuaikan berdasarkan dengan aturan *error-correction*. Respons sebenarnya dari *network* dikurangkan dengan respons target yang diinginkan untuk menghasilkan sinyal *error*. Kemudian sinyal *error* disebar ke arah yang berlawanan dengan arah koneksi sinapsis, oleh karena itu disebut *error back propagation*[6].

2.6 Sensor Precision Potensiometer Bourns 3590S-2-103L

Potensiometer adalah resistor dengan tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dan dapat diatur nilai tegangan atau nilai resistansinya.

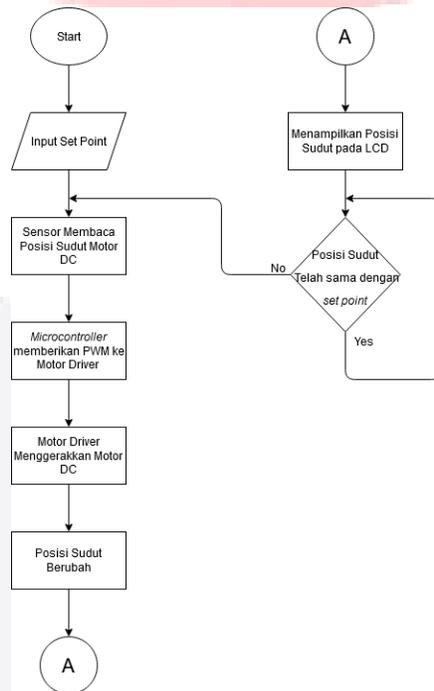
3. Perancangan Sistem

3.1. Perancangan Umum Sistem

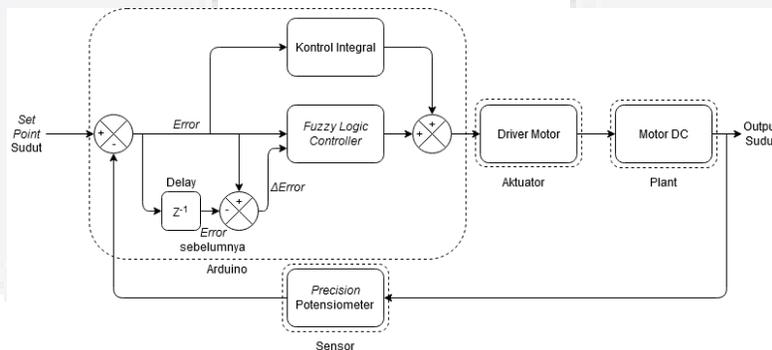
Perancangan umum sistem menjelaskan tentang perancangan utama seperti perancangan *hardware* dan *software* yang tepat untuk penelitian ini. Pada tahap awal, sistem akan diberikan *set point* berupa posisi sudut yang diinginkan. Kemudian data *set point* akan masuk ke *microcontroller* lalu data tersebut akan diolah dan *output* dari *microcontroller* akan menggerakkan motor DC yang terhubung dengan sensor potensiometer, sehingga motor DC akan bergerak menyesuaikan dengan *set point*.

Target sistem yang akan dirancang pada penelitian ini adalah sistem dapat mengatur sudut motor DC berdasarkan *set point* secara akurat dengan menggunakan sensor potensiometer dan FLC serta NNC sebagai metode kendali yang digunakan pada sistem ini.

Sistem ini menggunakan dua metode kendali, sehingga memiliki dua diagram blok dan memiliki satu *flowchart* yang sama seperti yang ditunjukkan pada Gambar 3 – 1. Diagram blok dengan menggunakan metode kendali FLC ditunjukkan pada Gambar 3 – 2.

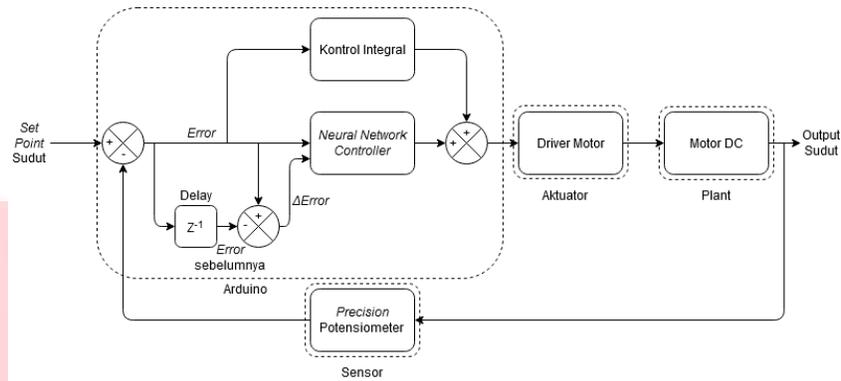


Gambar 3 - 1. Flowchart Sistem



Gambar 3 - 2. Diagram Blok FLC

Pada Gambar 3 – 2 terdapat kontrol integral yang berfungsi untuk menurunkan *error steady state*. Berdasarkan Gambar 3 – 3 dan *delay* memiliki fungsi untuk memanggil memori sebelumnya. Diagram blok dengan menggunakan metode kendali *NNC* ditunjukkan pada Gambar 3 – 3.

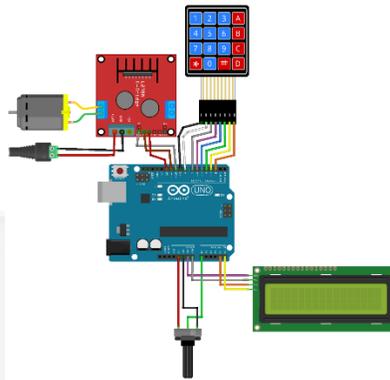


Gambar 3 - 3. Diagram Blok NNC

Perbedaan pada Gambar 3 – 2 dan Gambar 3 – 3 terletak pada kontroler sistemnya. Pada Gambar 3 – 2 menggunakan FLC sebagai kontroler sistemnya, sedangkan pada Gambar 3 – 3 menggunakan NNC sebagai kontroler sistemnya.

3.2. Perancangan *Hardware*

Diagram blok *hardware* dengan menggunakan FLC dapat dilihat pada Gambar 3 – 4.



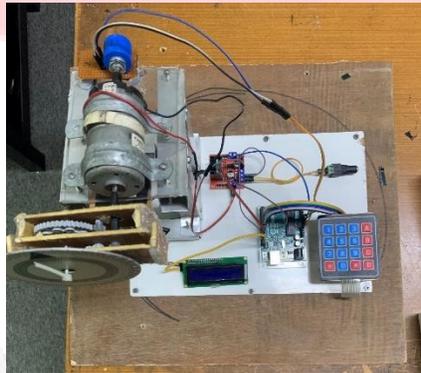
Gambar 3 - 4. Diagram Blok Hardware FLC

Berdasarkan Gambar 3 – 4, arduino uno akan menjalankan program yang telah dimasukkan ke dalam arduino tersebut. *Keypad* 4x4 digunakan sebagai *set point* untuk mengatur sudut yang kita inginkan. Arduino uno akan memproses sinyal masukan dari *keypad*, kemudian arduino memberikan sinyal ke *hardware* lainnya (selain *keypad*) untuk bekerja. LCD 16x2 digunakan untuk menampilkan *input* sudut yang diberikan dan *output* sudut yang terdeteksi oleh sensor. Aktuator yang digunakan adalah L298N *dual h-bridge DC motor driver* yang memiliki tegangan masukan berkisar 3.2V ~ 40V DC serta memiliki dua I/O (pada alat ini digunakan satu I/O). *Plant* yang digunakan adalah motor DC 12V 2600 rpm.

Komponen – komponen yang digunakan ditempatkan pada tripleks berukuran 30 cm x 20 cm, dan kayu dengan ukuran 41 cm x 38 cm. Komponen dan modul yang digunakan yaitu *microcontroller*, LCD 16x2, *keypad* 4x4, *driver motor* L298N, sensor *precision* potensiometer, dan motor DC. Perancangan mekanik pada penelitian dapat dilihat pada Gambar 3 – 5 dan Gambar 3 – 6.



Gambar 3 - 5. Arsitektur *Hardware* Tampak Depan



Gambar 3 - 6. Arsitektur *Hardware* Tampak Atas

Pada gambar di atas, motor diletakkan pada *stand* yang terbuat dari aluminium dengan ketebalan 2 mm. *Stand* tersebut berukuran Panjang x Lebar x Tinggi , 13 cm x 10 cm x 24 cm. Motor DC dipasangkan ke *gearbox* dengan rasio 1:10, karena sensor *precision* potensiometer memiliki 10 putaran untuk mencapai pembacaan maksimum.

Fungsi *hardware* pada penelitian ini adalah sebagai alat untuk memberikan *output* posisi sudut dari motor DC yang mampu bekerja dengan performa yang baik dan mampu menampilkan *output* secara analog maupun digital. Adapun fitur-fitur pada *hardware* adalah:

1. Motor DC dapat diatur sudutnya dengan menggunakan *keypad* 4x4.
2. Alat dikontrol menggunakan dua metode yang berbeda.

3.3. Perancangan *Software*

Pada penelitian ini digunakan *software* MATLAB untuk merancang sistem *Fuzzy Logic Controller* (FLC) dan *Neural Network Controller* (NNC) sebelum diterjemahkan ke dalam bahasa C pada pemrograman Arduino IDE. MATLAB merupakan *software* untuk implementasi dari algoritma, manipulasi matriks, pengolahan data, penempatan fungsi dan pembuatan antarmuka. Pada penelitian ini digunakan untuk merancang FLC dan NNC.

Arduino IDE merupakan *software* pemrograman untuk melakukan fungsi-fungsi tertentu dalam suatu *source code*. Pada penelitian ini digunakan untuk membuat *source code* untuk di-*upload* pada *microcontroller* Arduino Uno, sehingga Arduino Uno mampu untuk melakukan perintah yang telah dibuat.

4. Hasil dan Analisis

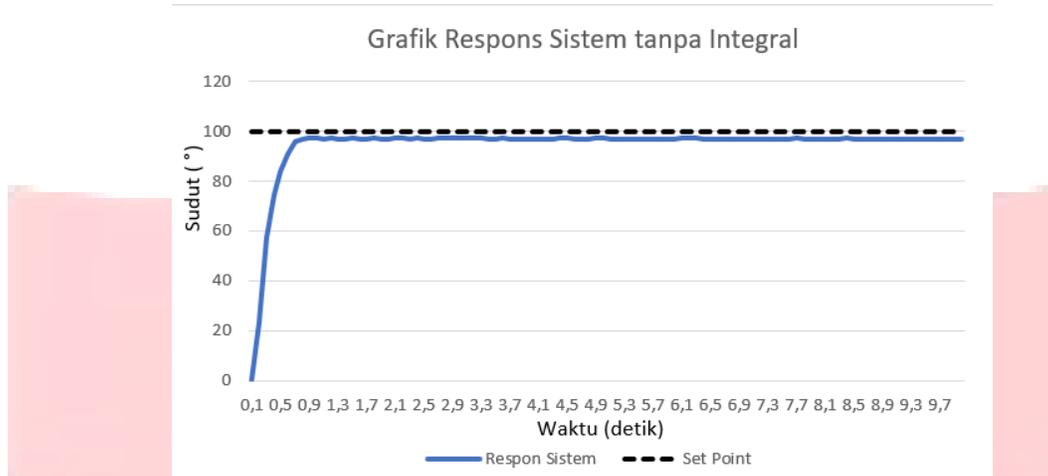
Pada bab ini akan dibahas mengenai pengujian dan analisis perangkat keras dengan menggunakan sistem yang telah dibuat.

4.1. Pengujian Perangkat Keras

Pengujian ini bertujuan untuk mengetahui respons sistem dan nilai *error* yang didapatkan ketika diberikan nilai *set point* yang kita inginkan. Pengujian ini terbagi atas tiga macam, yaitu pengujian sistem tanpa kontrol integral, pengujian sistem dengan kontrol integral, dan pengujian sistem ketika diberikan gangguan. Secara keseluruhan sistem menggunakan tambahan kontrol integral yang berfungsi untuk mengurangi *error steady state* dari respons sistem.

- a. Pengujian sistem tanpa kontrol integral

Pada pengujian ini, sistem dijalankan tanpa menggunakan integral dan diberikan *set point* sudut 100°. Bentuk grafik respons sistem tanpa menggunakan integral dengan *set point* sudut 100° dapat dilihat pada Gambar 4 – 1.

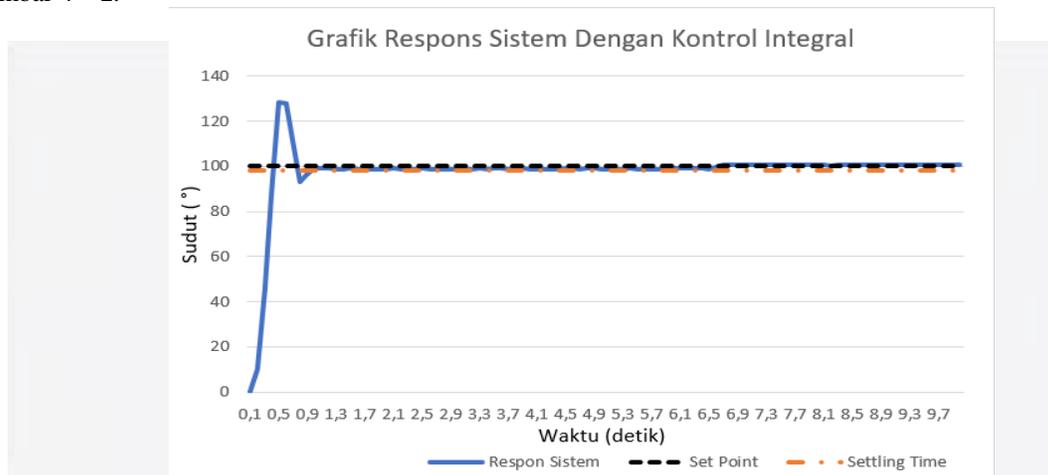


Gambar 4 - 1. Grafik Respons Sistem tanpa Integral *Set Point* 100°

Berdasarkan Gambar 4 – 1, sistem telah stabil disudut 97,13° dan membutuhkan waktu 0,9 detik untuk stabil. Sistem tersebut memiliki *error steady state* sebesar 2,87%.

b. Pengujian sistem dengan kontrol integral

Pada pengujian ini, sistem dijalankan dengan menggunakan integral dan diberikan *set point* sudut 100°. Bentuk grafik respons sistem dengan menggunakan integral dengan *set point* sudut 100° dapat dilihat pada Gambar 4 – 2.

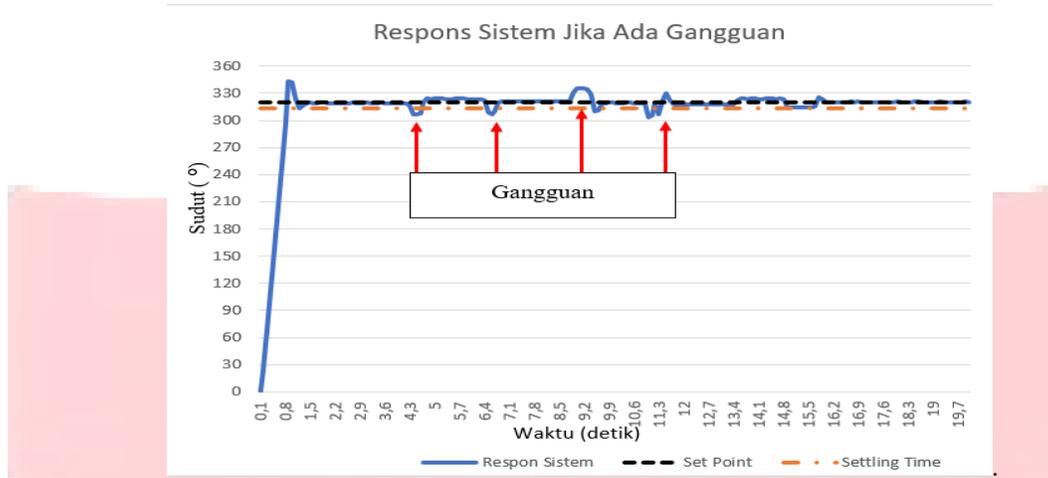


Gambar 4 - 2. Grafik Respons Sistem dengan Integral *Set Point* 100°

Pada Gambar 4 – 2, sistem telah stabil disudut 99,14°, nilai *overshoot* sebesar 28,32% pada waktu 0,5 detik, sistem mencapai keadaan tunak dalam waktu 0,9 detik, dan membutuhkan waktu 1,1 detik untuk stabil dengan *error steady state* 0,86%.

c. Pengujian sistem ketika diberikan gangguan

Pengujian ini bertujuan untuk mengetahui respons sistem apabila diberikan gangguan kepada sistem. Grafik dari respons sistem dapat dilihat pada Gambar 4 – 3.

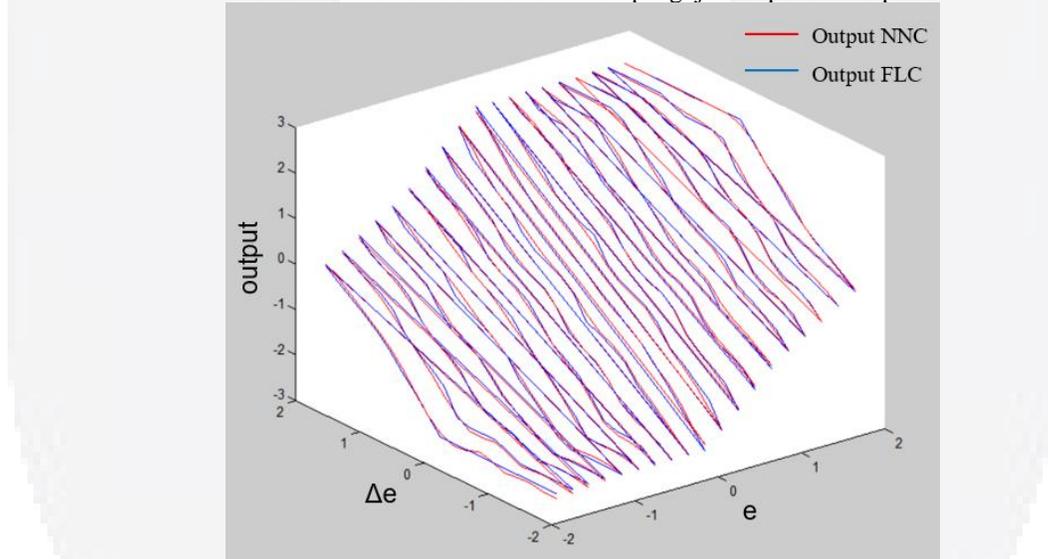


Gambar 4 - 3. Grafik Respons Sistem Jika diberikan Gangguan

Berdasarkan Gambar 4 – 3, saat diberikan *set point* disudut 320°, nilai *overshoot* sebesar 7,116% pada waktu 0,9 detik, sistem mencapai keadaan tunak dalam waktu 1,3 detik, sistem membutuhkan waktu 1,4 detik untuk mulai stabil disudut 319,22° dan *error steady state* sebesar 0,24%. Ketika diberi gangguan sebanyak empat kali sistem mampu kembali menuju *set point* dengan sudut 319,92° dan *error steady state* sebesar 0,03%.

4.2. Pengujian Simulasi Menggunakan 16 Neuron

Pengujian ini bertujuan untuk mengetahui nilai *error* sistem dengan menggunakan sistem kontrol *neural network* yang menggunakan satu *hidden layer* dan 16 neuron. Pengujian ini menggunakan 361 data latih yang didapatkan dengan cara menaikkan sudut setiap 20° untuk *error* dan $\Delta error$. Pengujian ini dilakukan dengan cara pengambilan nilai secara acak untuk nilai *error* dan $\Delta error$. Hasil pengujian dapat dilihat pada Gambar 4 – 4.



Gambar 4 - 4. Hasil Training Neural Network Controller Menggunakan 16 Neuron pada Hidden Layer

Berdasarkan Gambar 4 – 4, hasil *training* NNC bekerja dengan menghasilkan *error steady state* rata-rata 1,40%. *Training* NNC selesai dilakukan dengan nilai *epoch* 663090 dan nilai MSE 0,00345 dengan nilai presisi 0,0035. Setelah dilakukan *training*, NNC akan dilakukan uji coba dengan memasukkan salah satu data *training* dari FLC, kemudian akan dilihat hasil dari NNC yang seharusnya akan sama atau mendekati hasil dari FLC.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang sudah dibuat dan diamati, didapatkan beberapa kesimpulan sebagai berikut:

1. Alat sudah direalisasikan dan dapat bekerja sesuai fungsi yang diinginkan.
2. Pengambilan data *training* untuk NNC dilakukan sebanyak empat kali. Pengambilan data pertama dilakukan setiap kenaikan sudut 12°, pengambilan data kedua dilakukan setiap kenaikan sudut 15°, pengambilan data ketiga dilakukan setiap kenaikan sudut 18°, dan pengambilan data keempat dilakukan setiap kenaikan sudut

- 20°. Hal ini dilakukan untuk melihat perbedaan respons sistem dan *error steady state* yang dihasilkan dari *training* NNC.
3. NNC memiliki respons sistem yang mendekati respons sistem yang dihasilkan dengan FLC dengan *error steady state* rata-rata <5%.
 4. Sistem FLC jika diberikan kontrol integral mampu memberikan *error steady state* <2%, bahkan ketika diberikan gangguan sistem mampu kembali menuju *set point* yang telah ditentukan dan tetap memiliki *error steady state* <2%.
 5. Hasil *training* NNC memiliki respons yang mendekati respons sistem FLC.
 6. Semakin banyak neuron yang digunakan pada *hidden layer* semakin bagus juga hasil dari *training* dan pengujian NNC. Namun, untuk melakukannya membutuhkan waktu yang lama. Pada sistem ini direkomendasikan untuk menggunakan 16 neuron atau lebih.

REFERENSI

- [1]. Anonimus. (2015, Maret 01). Diambil kembali dari <http://weeq-blajar.blogspot.com/2015/03/kelas-xi-turunan-dan-integral-vektor.html>, pada 29 November 2019, pukul 10:40 WIB.
- [2]. Dwi, Novianto. (2010, April 29). Diambil kembali dari <https://anto12.wordpress.com/2010/04/29/pengertian-sistem-kontrolkendali>, diakses pada 17 Oktober 2019, pukul 19:10 WIB.
- [3]. Fahmizal. (2012, September 30). Diambil kembali dari <http://fahmizaleeits.wordpress.com/category/kuliah-kontrol/pid-kontrol/>, diakses pada 17 Oktober 2019, pukul 19:53 WIB.
- [4]. Martin T. Hagan, Howard B. Demuth, Mark Beale. *Neural Network Design*. China Machine Press, 2002. ISBN : 9780971732100
- [5]. Ogata, Katsuhiko. 1991. *Teknik Kontrol Automatik*, Jakarta:Erlangga. ISBN : 979-411-467-7.
- [6]. Pinjare, S. L. & M., Arun Kumar. *Implementation of Neural Network Back Propagation Training Algorithm on FPGA*. International Journal of Computer Applications, Vol 52, No.6, 2012.
- [7]. Ramadhan, Fareza Rizky. 2017. *Perancangan dan Implementasi Kontrol Posisi Robot Bawah Air Menggunakan Metode Fuzzy Logic Control* [skripsi]. Bandung (ID): Telkom University.