

## Tuning Parameter pada Pengendali Logika Fuzzy menggunakan Algoritma *Ant Colony Optimization*

### *Parameter Tuning for a Fuzzy Logic Controller using the Ant Colony Optimization Algorithm*

Adriana Fanggidae\*

Program studi Ilmu Komputer, Fakultas Sains dan Teknik, Universitas Nusa Cendana  
Jl. Adisucipto, Lasiana, Kelapa Lima, Kupang, Nusa Tenggara Timur 85228.

\*Email: [adrianafanggidae@staf.undana.ac.id](mailto:adrianafanggidae@staf.undana.ac.id)

**Abstrak** – Tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma Ant Colony Optimization (ACO) untuk melakukan tuning parameter pada pengendali logika fuzzy atau *Fuzzy Logic Controller* (FLC). FLC tersebut digunakan untuk mengendalikan ketinggian air dari tangki proses. Masukan dan keluaran fuzzy terdiri dari tujuh fungsi keanggotaan, yaitu positif besar (PB), positif menengah (PM) dan positif kecil (PS), zero (Z), negatif kecil (NS), negatif menengah (NM) dan negatif besar (NB). Pertama-tama, dicari parameter FLC awal, lalu dibangkitkan suatu graph dimana nilai-nilai parameter FLC ditentukan dalam rentang nilai antara 0 hingga 1,5 kali dari nilai parameter awal. Lalu diimplementasikan algoritma ACO untuk memperbaiki nilai parameter FLC tersebut agar diperoleh performansi yang lebih baik. Performansi pengendali yang diharapkan adalah meminimalkan lonjakan maksimum (*overshoot*) dan waktu naik (*rise time*). Sistem ini diimplementasikan menggunakan program labVIEW. Data ketinggian air diperoleh menggunakan sensor potensiometer. Keluaran dari FLC terhubung dengan motor stepper untuk mengatur debit masukan air ke tangki proses. Dari hasil pengujian telah berhasil diperoleh *overshoot* dan *risetime* yang kecil, sebagai contoh, untuk setpoint 8, performansi keluaran sistem memiliki *overshoot* 2.5% dan *rise time* 8909 ms. Algoritma ACO berhasil meningkatkan performansi sistem dibandingkan performansi sistem jika menggunakan parameter awal. Peningkatan performansi ini dikarenakan algoritma ACO bertindak sebagai algoritma pencarian lokal (*local search*) yang akan mencari performansi sistem yang lebih baik disekitar nilai parameter awalnya. Penelitian ini berhasil menunjukkan bahwa algoritma ACO dapat digunakan untuk melakukan tuning dari parameter FLC.

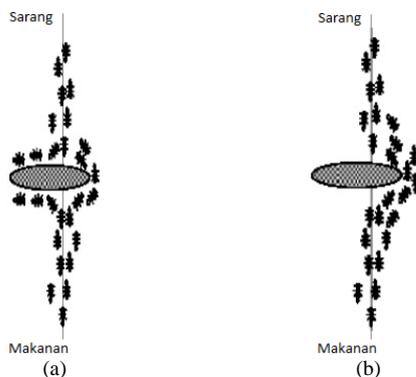
**Kata Kunci** : *Ant colony optimization*, pencarian lokal, pengendalian ketinggian air, pengendali logika fuzzy, tuning parameter

**Abstract** - The purpose of this study is to implement the Ant Colony Optimization (ACO) algorithm for tuning parameters in a fuzzy logic controller or Fuzzy Logic Controller (FLC). The FLC is used to control the water level of the process tank. Fuzzy input and output consists of seven membership functions, namely large positive (PB), medium positive (PM) and small positive (PS), zero (Z), small negative (NS), medium negative (NM) and large negative (NB). First, the initial FLC parameter is searched, then a graph is generated where the values of the FLC parameter are determined in the range of values between 0 and 1.5 times the initial parameter value. Then the ACO algorithm is implemented to improve the value of the FLC parameters in order to obtain better performance. The expected controller performance is to minimize the overshoot and rise time. This system is implemented using the LabVIEW program. Water level data is obtained using a potentiometer sensor. The output from the FLC is connected to the stepper motor to regulate the discharge of water input to the process tank. From the test results we have obtained a small overshoot and research, for example, for setpoint 8, the system output performance has an overshoot of 2.5% and a rise time of 8909 ms. ACO algorithm succeeded in increasing system performance compared to system performance if using initial parameters. This increase in performance is due to the ACO algorithm acting as a local search algorithm which will look for better system performance around its initial parameter values. This research successfully demonstrated that the ACO algorithm can be used to perform tuning of FLC parameters.

**Keywords** : *Ant colony optimization*, local search, water level control, fuzzy logic controller, parameter tuning

## I. PENDAHULUAN

Pengendali logika Fuzzy atau *Fuzzy Logic Controller* (FLC) telah menjadi salah satu bidang penelitian yang paling aktif. Beberapa penelitian telah menunjukkan bahwa FLC memberikan performansi pengendalian yang lebih baik dibandingkan metode pengendalian konvensional [1-3]. Tetapi FLC dan parameternya harus dirancang khusus untuk setiap sistem yang hendak dikendalikan. Perancangan dan penyesuaian ini biasanya dilakukan secara manual atau dengan metode *trial and error*. Tetapi karena penyesuaian parameter dengan cara manual atau *trial and error* tersebut menghabiskan waktu yang banyak, maka salah satu alternatifnya adalah menggunakan teknik seperti Algoritma Genetik (GA) [4], Particle Swarm Optimization (PSO) [5] ataupun Ant Colony Optimization (ACO) [6]. Pendekatan menggunakan ACO menjadi salah satu metode yang digunakan pada penelitian ini. Algoritma *Ant Colony Optimization* (ACO) diinspirasi dari perilaku kelompok semut dalam mencari dan mengumpulkan makanan. Prinsip dari algoritma ini ditunjukkan pada **Gambar 1**. Algoritma ini ditujukan untuk menyelesaikan masalah optimasi diskrit [7]. Algoritma ACO telah berhasil diimplementasikan pada berbagai masalah optimasi diskrit, seperti perutean kendaraan [8], penjadwalan pekerjaan [9], ataupun masalah *travelling-salesman problem* (TSP) [10]. Penelitian ini mengimplementasikan FLC untuk mengendalikan tangki proses, dimana parameter FLC akan detuning menggunakan algoritma ACO. Pengendalian tangki proses adalah sistem yang sederhana dan mudah dipahami, serta tidak stabil dalam kasus lup terbuka.



**Gambar 1.** Prinsip dari algoritma ACO : (a) Kelompok semut akan mencari jalan menuju makanan dan kembali ke sarang. (b) rute yang terpendek akan memiliki lebih banyak pheromone karena pada rute terpendek, semut akan lebih banyak bolak balik antara sarang dan makanan

Terdapat beberapa publikasi mengenai penelitian tuning parameter FLC. Castilo [6] membahas penelitian mengenai tuning parameter Fuzzy menggunakan ACO, tetapi pengujian yang dilakukan masih menggunakan simulasi. Castilo [11] melakukan penelitian mengenai tuning parameter Fuzzy secara dinamik menggunakan ACO untuk implementasi robot mobil. Tetapi paper tersebut belum membahas proses implementasinya pada robot mobil. Mitra [12] melakukan penelitian mengenai pengendali fuzzy supervised menggunakan algoritma ACO, tetapi belum menguji hasilnyanya pada sistem nyata. Kaliannan [13] mengimplementasikan ACO untuk tuning parameter pengontrol Proportional-Integral-Derivative (PID), tetapi pengujiannya juga masih dilakukan secara simulasi. Sejauh yang diketahui oleh penulis, belum ada penelitian sebelumnya yang menyajikan hasil tuning parameter FLC menggunakan algoritma ACO yang pengujiannya dilakukan secara eksperimen pada sistem nyata.

Tujuan dari penelitian ini adalah mengimplementasikan algoritma ACO untuk tuning parameter FLC pada pengendalian tinggi air pada tangki proses. Pengujian akan dilakukan pada alat nyata. Metode yang digunakan adalah pertama-tama mencari parameter FLC awal. Kemudian dibuat suatu graph yang berisi nilai-nilai parameter FLC yang memiliki rentang 0 hingga 1,5 kali dari nilai parameter awal. Graph ini dibuat untuk mentransformasikan masalah optimasi parameter-parameter pengontrol Fuzzy kedalam bentuk permasalahan optimasi kombinatorial, sehingga masalah tuning parameter Fuzzy ini dapat dilakukan oleh algoritma ACO. Pada graph tersebut diimplementasikan algoritma ACO untuk memperbaiki nilai parameter FLC agar diperoleh performansi sistem yang lebih baik. Performansi sistem yang diharapkan adalah lonjakan maksimum (*overshoot*) dan waktu naik (*rise time*) yang minimal. Sistem ini diimplementasikan menggunakan program labVIEW, dengan sensor masukan menggunakan potensiometer dan aktuator menggunakan motor stepper untuk mengatur debit masukan air ke tangki proses. Pada metode yang diajukan ini, algoritma ACO bertindak sebagai algoritma *local search* yang akan mencari performansi sistem yang lebih baik disekitar nilai parameter awalnya. Penelitian ini akan menunjukkan bahwa algoritma ACO dapat digunakan untuk melakukan tuning dari parameter FLC.

**I. METODE**

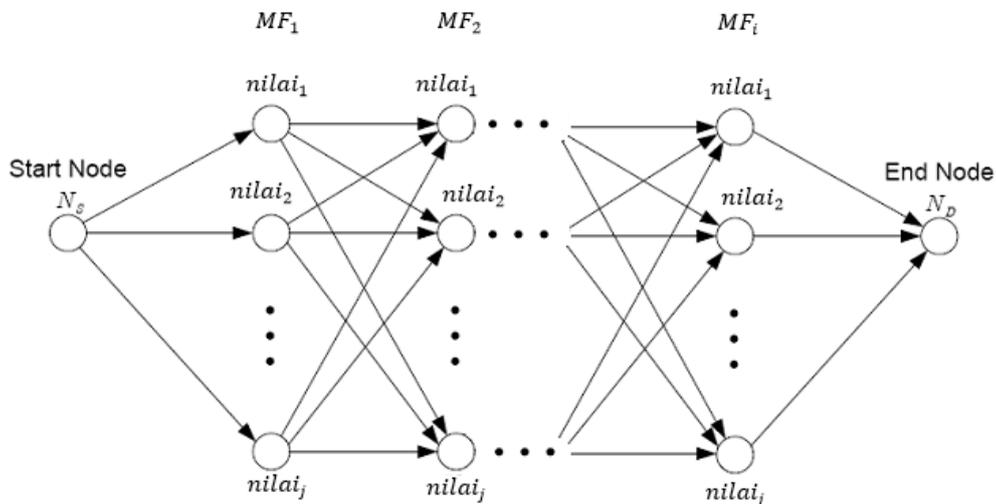
Masukan dan keluaran dari FLC yang digunakan terdiri dari tujuh fungsi keanggotaan, yaitu positif besar (PB), positif menengah (PM) dan positif kecil (PS), zero (Z), negatif kecil (NS), negatif menengah (NM) dan negatif besar (NB). Nilai-nilai dari parameter fungsi keanggotaan tersebut akan dituning menggunakan algoritma ACO. Metode untuk melakukan tuning parameter FLC menggunakan algoritma ACO ditunjukkan pada **Gambar 2**. Pertama-tama dibangkitkan fungsi keanggotaan dan parameter awalnya berdasarkan rentang data masukan dan keluaran FLC. Kemudian diimplementasikan algoritma ACO untuk melakukan *local search* disekitar nilai parameter awal FLC tersebut agar diperoleh performansi sistem yang lebih baik

```

Procedure Tuning_Fuzzy_ACO
    MakingDefaultFuzzyRuleBase
    AntColonyOptimization
End Tuning_Fuzzy_ACO
    
```

**Gambar 2.** Metode tuning parameter FLC menggunakan algoritma ACO

Berdasarkan nilai parameter FLC awal tersebut, lalu dibuat graph yang pada setiap cabanganya berisi alternative nilai-nilai parameter FLC yang memiliki rentang 0 hingga 1,5 kali dari nilai parameter awal. Ilustrasi graph yang dibuat ditunjukkan pada **Gambar 3**. Graph ini digunakan untuk mentransformasikan masalah optimasi parameter-parameter fungsi keanggotaan Fuzzy kedalam bentuk permasalahan optimasi kombinatorial. Melalui graph inilah algoritma ACO akan diterapkan. Dalam percobaan ini, jumlah *j* yang digunakan adalah 500.



**Gambar 3.** Representasi graph permasalahan tuning parameter Fuzzy

Semut-semut akan meletakkan *pheromone* pada setiap jalur graph yang dilaluinya. *Pheromone* akan diperbaharui /di-update melalui dua cara, yaitu update *pheromone* local, dan update *pheromone* global. Pada update *pheromone* lokal, jika semut telah selesai melalui suatu jalur dari titik awal ke tujuan, maka *pheromone* akan di-update menggunakan persamaan (1).

$$\tau_{ij}(t) = \tau_{ij}(t - 1) + C_{nn} \tag{1}$$

Dengan  $C_{nn} = \frac{1}{n \times L_{nn}}$ .

Variabel  $C_{nn}$  adalah variabel peng-update-an *pheromone* secara umum. *n* adalah jumlah jalur dan  $L_{nn}$  adalah nilai fitness dari performansi sistem jika menggunakan parameter-parameter yang diberikan pada jalur tersebut.

Pada update *pheromone* global, ada update *pheromone* positif (persamaan 2) dan ada update *pheromone* negative (persamaan 3). Persamaan yang digunakan adalah sebagai berikut.

$$\tau_{ij}^{best}(t) = \tau_{ij}^{best}(t - 1) + \frac{\rho}{L_{best}} \tag{2}$$

$$\tau_{ij}^{worst}(t) = \tau_{ij}^{worst}(t - 1) - \frac{0.3\rho}{L_{worst}} \tag{3}$$

Dimana  $\tau_{ij}^{best}$  dan  $\tau_{ij}^{worst}$  adalah *pheromone* dari masing-masing jalur yang dilalui semut dalam *tour* dengan nilai fitness terendah ( $C_{best}$ ) dan tertinggi ( $C_{worst}$ ) dalam satu iterasi. *Pheromone* dari masing-masing jalur yang membangun *best tour* akan terus bertambah, sedangkan *pheromone* dari jalur yang membangun *worst tour* akan terus berkurang selama iterasi.

Setelah satu iterasi, beberapa *pheromone* akan menguap. Penguapan *pheromone* pada persamaan (4) akan membuat algoritma ACO untuk melupakan histori sebelumnya, sehingga ACO dapat lebih terbuka untuk melakukan pelacakan pada arah yang baru tanpa akan terjebak pada *local minimal*.

$$\tau_{ij}(t) = \tau_{ij}(t)^\lambda + \Delta \quad (4)$$

Dimana  $\lambda$  adalah konstanta penguapan dan  $\Delta$  adalah jumlah dari persamaan (2) dan (3).

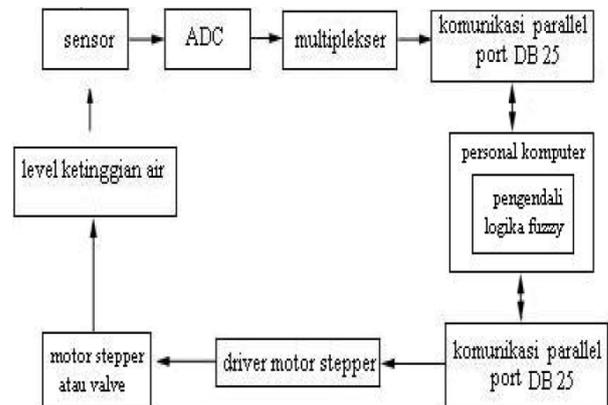
Performansi sistem yang hendak diminim adalah jumlah dari persamaan alkan adalah nilai *overshoot* dan waktu *risetime*. Fitness yang digunakan untuk optimasi parameter fuzzy dengan ACO pada penelitian ini adalah *Integral Absolute Error* (IAE) dan *Integral Squared Error* (ISE).

## II. HASIL DAN DISKUSI

Diagram blok yang dirancang pada penelitian ini ditunjukkan pada **Gambar 4**. Urutan kerjanya adalah sebagai berikut.

1. Sensor potensiometer menangkap informasi ketinggian air berdasarkan perubahan tegangan, kemudian mengirimkan sinyal analog ke *Analog to Digital Converter* (ADC)
2. Jika telah diterima oleh ADC, maka sinyal analog dari sensor diubah menjadi data digital atau biner 8 bit, lalu keluaran dari ADC akan diproses oleh rangkaian multiplexer.
3. Dalam rangkaian multiplexer, sinyal dari ADC diproses dan kemudian dikeluarkan dalam satu saluran, data ini kemudian dikirim ke komputer melalui komunikasi paralel menggunakan konektor DB25 pada komputer.
4. Data yang telah memasuki komputer kemudian diproses menggunakan perangkat lunak LabVIEW untuk menerapkan kontrol logika fuzzy, setelah itu data yang diolah kemudian dikeluarkan kembali ke konektor DB25, yang kemudian digunakan untuk mengendalikan motor stepper untuk mengontrol buka dan tutup sistem keran sehingga ketinggian air dapat dikendalikan. Data yang telah masuk ke komputer, kemudian diolah dan diproses menggunakan software LabVIEW untuk mengimplementasikan kendali logika fuzzy, setelah itu data yang telah diproses kemudian dikeluarkan kembali pada connector DB 25, yang kemudian digunakan untuk mengendalikan motor stepper untuk

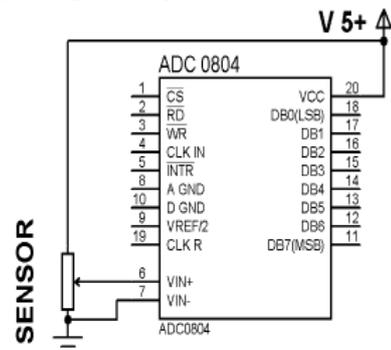
mengendalikan sistem buka dan tutup keran sehingga level ketinggian air dapat dikendalikan.



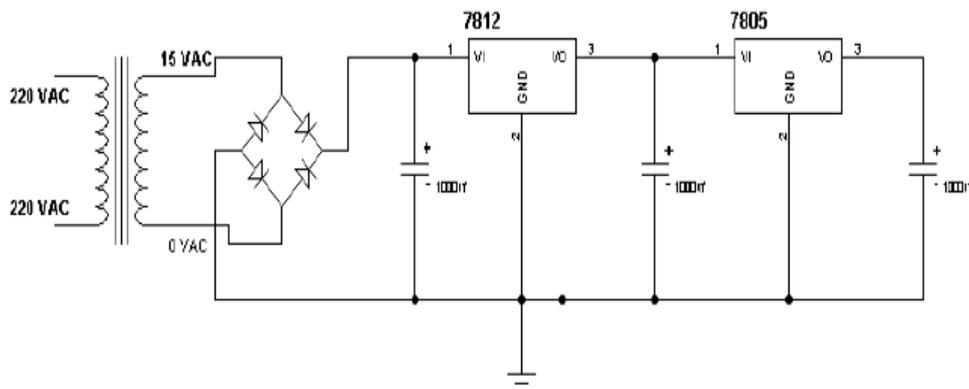
Gambar 4. Blok diagram sistem

Pada perancangan perangkat keras, telah dibuat beberapa sub-rangkaian, yaitu sebagai berikut.

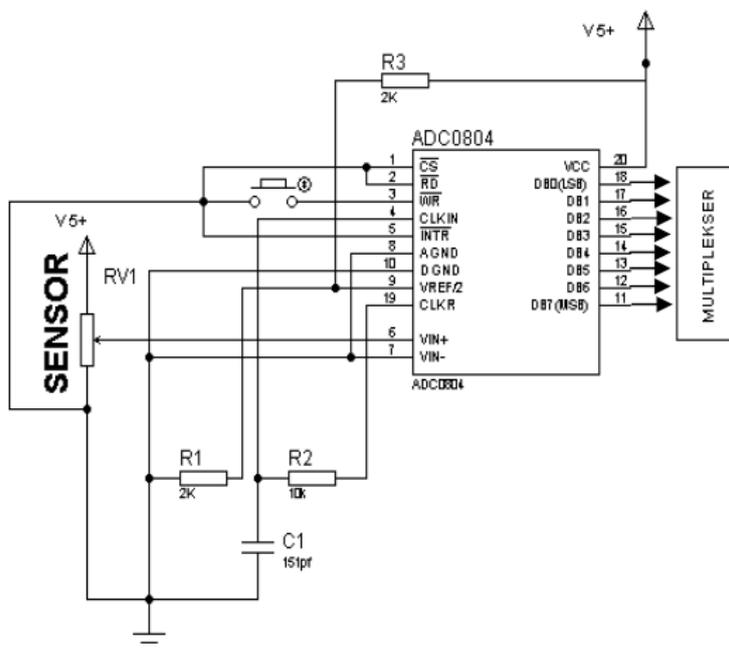
- a. Rangkaian sensor level ketinggian menggunakan potensiometer yang ditunjukkan pada **Gambar 5**.
- b. Rangkaian catu daya sebagai supply tegangan, yang ditunjukkan pada **Gambar 6**.
- c. Rangkaian ADC 0804 untuk mengubah data analog dari sensor menjadi data digital, yang ditunjukkan pada **Gambar 7**. Karena menggunakan ADC 8 bit, maka diperoleh resolusi tegangan masukan sebesar 19,6 mV.
- d. Rangkaian multiplexer untuk mengubah data parallel menjadi serial yang akan dikeluarkan pada satu output, yang ditunjukkan pada **Gambar 8**.
- e. Rangkaian driver motor stepper untuk mengendalikan bukaan keran, yang ditunjukkan pada **Gambar 9**.
- f. Penggunaan konektor DB25 sebagai interface antara perangkat keras dengan komputer, yang ditunjukkan pada **Gambar 8** dan **9**.



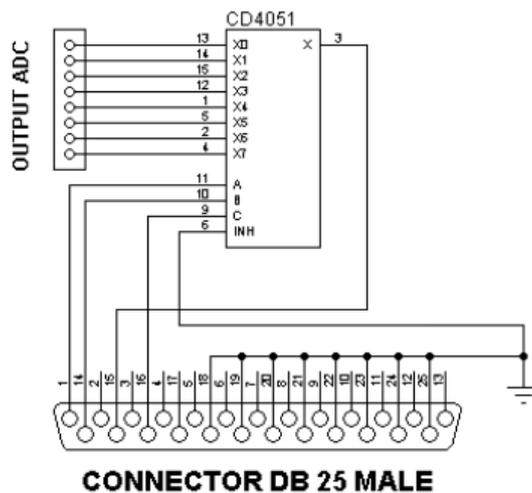
Gambar 5. Rangkaian sensor potensiometer



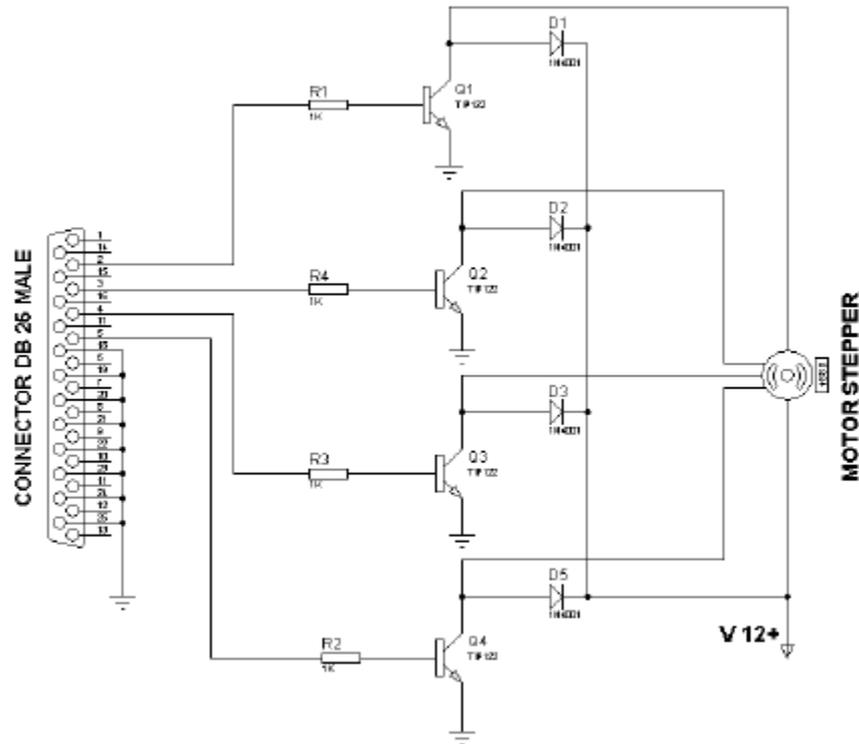
Gambar 6. Rangkaian catu daya



Gambar 7. Rangkaian ADC dengan sensor

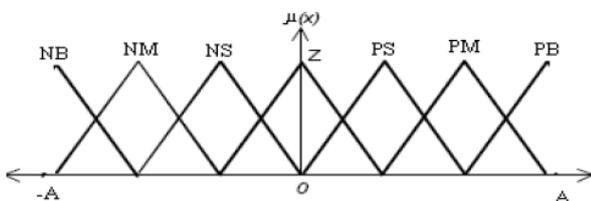


Gambar 8. Rangkaian Multiplexer dengan port parallel DB25

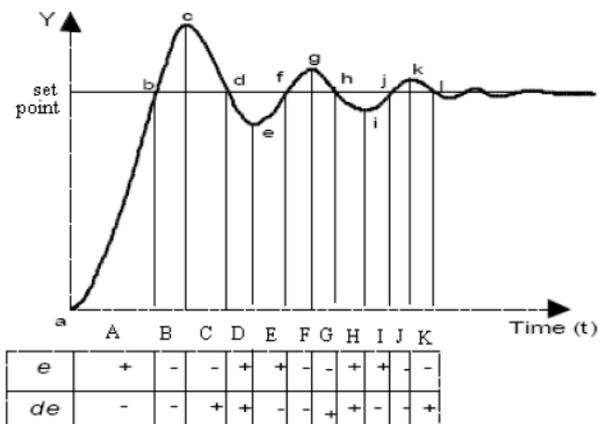


Gambar 9. Rangkaian driver motor stepper dengan port parallel DB25

Selanjutnya akan dilakukan perancangan kaidah fuzzy. Perancangan kaidah fuzzy ini seperti yang dilakukan pada [14]. Basis kaidah yang digunakan pada logika fuzzy ini 7 variabel linguistik, yaitu positif besar (PB), positif menengah (PM), positif kecil (PS), nol (Z), negatif besar (NB), negatif menengah (NM), negatif kecil (NS). Seperti terlihat pada Gambar 10. Masukan yang digunakan adalah  $e$  dan  $\Delta e$ . Dengan mengacu pada tanggapan (*step respon*) sistem lup tertutup seperti Gambar 11, maka dapat dirancang kaidah kendali fuzzy seperti yang ditunjukkan pada Tabel I. Kaidah 1 pada Tabel I menunjukkan keadaan pada titik a di Gambar 11. Titik a menunjukkan keadaan error ( $e$ ) yang besar (PB), maka aksi yang harus dilakukan oleh sinyal kontrol ( $U$ ) adalah melakukan bukaan valve yang besar (PB) agar level ketinggian air segera naik. Pada titik c ketinggian air telah melebihi setpoint (error bernilai negative), maka error pada titik c adalah NB. Maka aksi yang harus dilakukan oleh sinyal kontrol adalah menutup rapat (NB) keran agar air segera turun.



Gambar 10. Fungsi keanggotaan Fuzzy yang digunakan



Gambar 11. Tanggapan (*step respon*) sistem lup tertutup yang diharapkan

Tabel I. Kaidah kendali Fuzzy yang dirancang untuk memperoleh *step respon* sistem lup tertutup seperti pada Gambar 11

Kaidah	$e$	$\Delta e$	$U$	Referensi	Fungsi
1	PB	Z	PB	Titik a	Perpendek <i>rise time</i>
2	PM	Z	PM	Titik e	Perpendek <i>rise time</i>
3	PS	Z	PS	Titik l	Perpendek <i>rise time</i>
4	Z	NB	NB	Titik b	Kurangkan <i>overshoot</i>
5	Z	NM	NM	Titik f	Kurangkan <i>overshoot</i>
6	Z	NS	NS	Titik j	Kurangkan <i>overshoot</i>
7	NB	Z	NB	Titik c	Kurangkan <i>overshoot</i>
8	NM	Z	NM	Titik g	Kurangkan <i>overshoot</i>
9	NS	Z	NS	Titik k	Kurangkan <i>overshoot</i>
10	Z	PB	PB	Titik d	Perkecil osilasi
11	Z	PM	PM	Titik h	Perkecil osilasi
12	Z	PS	PS	Titik i	Perkecil osilasi
13	Z	Z	Z	Titik set	

Agar pengendali *fuzzy* yang dirancang berlaku untuk seluruh masukan sistem, maka kaidah-kaidah kendali yang disusun pada **Tabel I** dilengkapi dengan kaidah-kaidah lainnya hingga jumlah kaidah kendali sama dengan jumlah kombinasi dari variabel linguistik *error* (*e*) dan turunan error ( $\Delta e$ ) yang tersedia. Kaidah lengkap yang dirancang ditunjukkan pada **Tabel 2**.

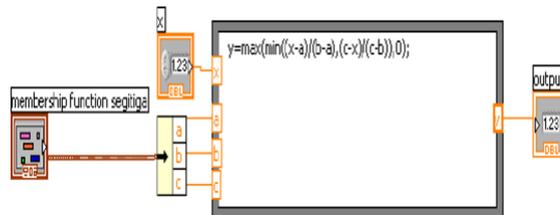
**Table II.** Kaidah kendali fuzzy yang digunakan dengan masukan tujuh membership function

$\Delta e$ <i>e</i>	NB	NM	NS	Z	PS	PM	PB
PB	Z	PS	PM	PB	PB	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PS	NM	NS	Z	PS	PM	PB	PB
Z	NB	NM	NS	Z	PS	PM	PB
NS	NB	NB	NM	NS	Z	PS	PM
NM	NB	NB	BN	NM	NS	Z	PS
NB	NB	NB	BN	BN	NM	NS	Z

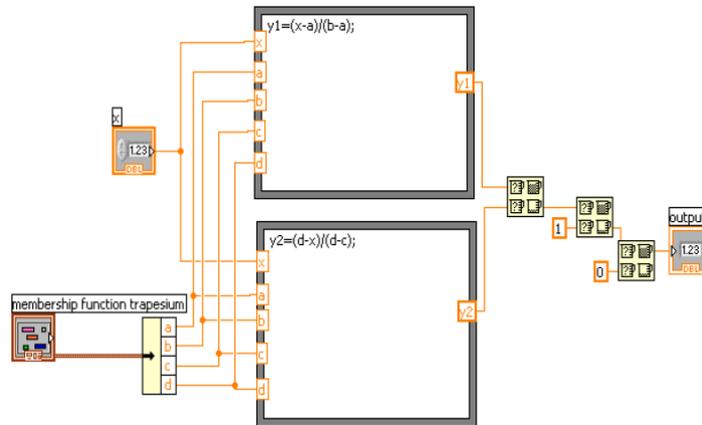
Pada penelitian ini telah dirancang beberapa sub-program pengendali fuzzy. Beberapa program yang dirancang adalah sebagai berikut.

1. Perancangan program Triangel.vi  
Program Triangel.vi berfungsi untuk menghitung keluaran dari fungsi keanggotaan segitiga pada logika fuzzy. Tampilan block diagram dari Triangle.vi ditunjukkan pada **Gambar 12**.
2. Perancangan program Trapezium.vi  
Program Trapezium.vi berfungsi untuk menghitung keluaran dari fungsi keanggotaan trapesium pada logika fuzzy. Tampilan block diagram dari Trapezium.vi ditunjukkan pada **Gambar 13**.
3. Perancangan program Input array.vi  
Program Input array.vi digunakan untuk membatasi besarnya nilai dan interval inputan pada fungsi keanggotaan segitiga dan trapesium. **Gambar 14** menunjukkan block diagram dari input array.vi.
4. Perancangan program Tangki.vi  
Perancangan program tangki.vi digunakan untuk mengamati perubahan level ketinggian air minimum 0 cm dan maksimum 15 cm. Data digital dari output ADC akan menginformasikan perubahan level ketinggian air pada tangki proses. Data-data tersebut diolah pada program tangki.vi untuk menampilkan level ketinggian air pada tangki proses. **Gambar 15** menunjukkan block diagram dari program tangki.vi

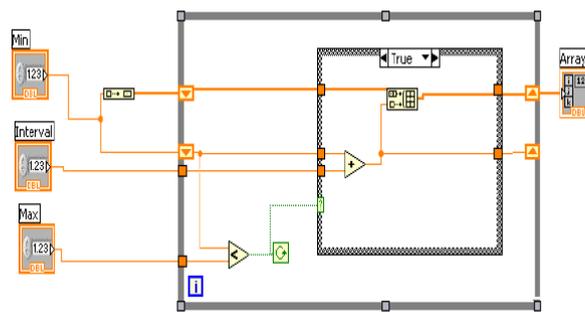
5. Perancangan program Pengaman.vi  
Program pengaman.vi digunakan sebagai sakelar. Prinsip tersebut memanfaatkan status 4 dan status 5 pada pin port parallel DB 25, yaitu pin 13 dan pin 12, jika status 4 aktif maka putaran keran akan mati hal ini menandakan bahwa keran telah membuka penuh sedangkan jika status 5 aktif maka putaran keran akan mati hal ini menandakan keran telah menutup penuh, **Gambar 16** menunjukkan block diagram dari program pengaman.vi
6. Perancangan program multiplekser.vi  
program Multiplekser.vi digunakan untuk menghitung nilai keluaran dari rangkaian multiplekser agar sesuai dengan keluaran dari ADC. Block diagram dari program multiplekser.vi ditunjukkan pada **Gambar 17**.
7. Perancangan program motor stepper.vi  
Program motor stepper.vi digunakan untuk mengendalikan putaran motor stepper supaya dapat memutar ke kanan, ke kiri dan berhenti. Block diagram dari program motor stepper.vi ditunjukkan pada **Gambar 18**.
8. Perancangan program fuzzy.vi  
Program fuzzy.vi berfungsi menggabungkan dan merangkai program triangel.vi, trapezium.vi dan input array.vi sehingga dapat merealisasikan kendali logika fuzzy dalam perangkat lunak (software), yang berisi proses fuzzifikasi, basis kaidah (rule base), mekanisme inferensi (*inference mechanism*) dan defuzzifikasi. Perancangan program pengontrol fuzzy.vi  
Program pengontrol fuzzy.vi digunakan untuk mengendalikan level ketinggian air dari plant sehingga level ketinggian air tersebut sesuai dengan level ketinggian yang diharapkan (setpoint). Perancangan program dilakukan dengan menggabungkan beberapa sub vi, yaitu triangel.vi, trapezium.vi, input array.vi, motor stepper.vi, tangki.vi, multiplekser.vi dan fuzzy.vi sehingga menjadi suatu kesatuan program pengontrol fuzzy.vi yang akan mengendalikan plant



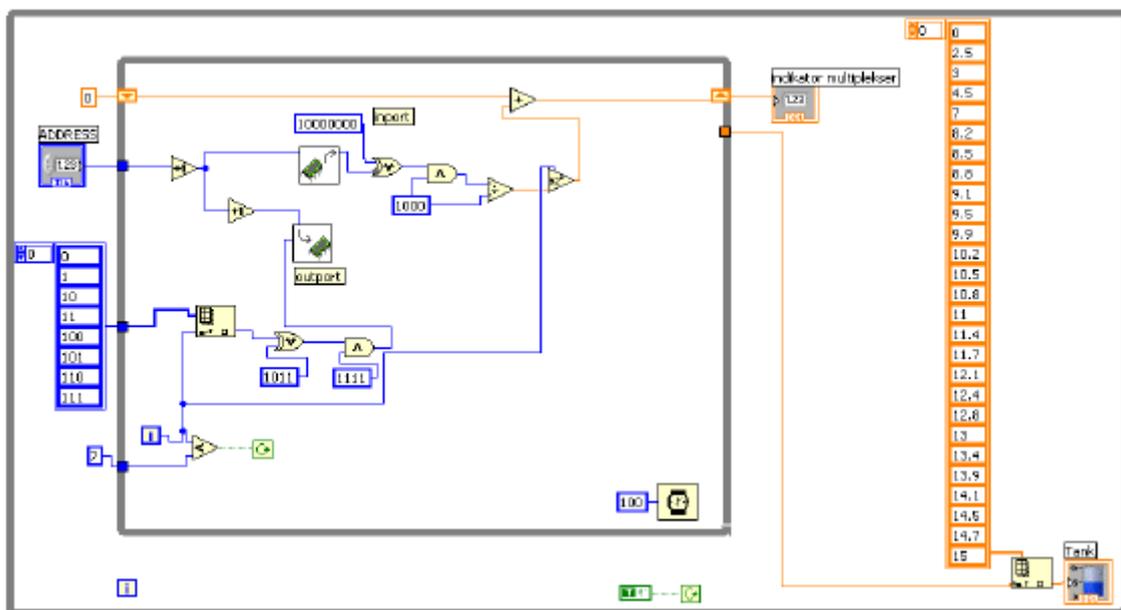
Gambar 12. Block diagram program triangle.vi



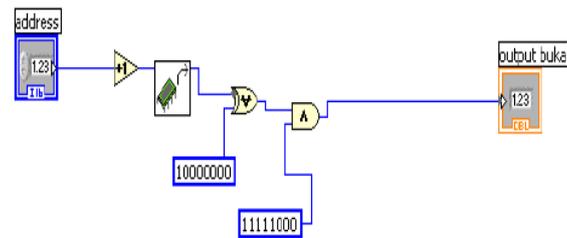
Gambar 13. Block diagram program trapezium.vi



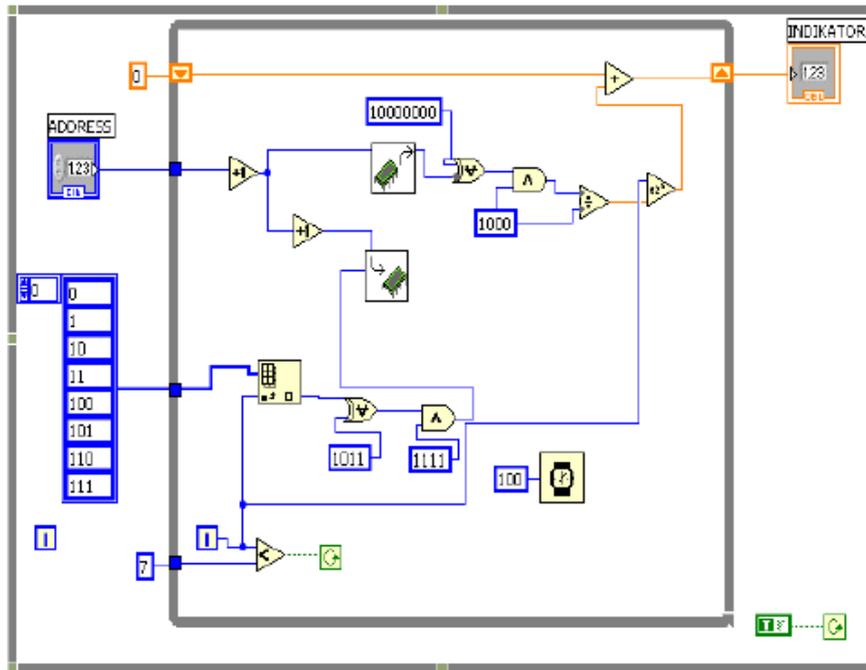
Gambar 14. Block diagram input array.vi



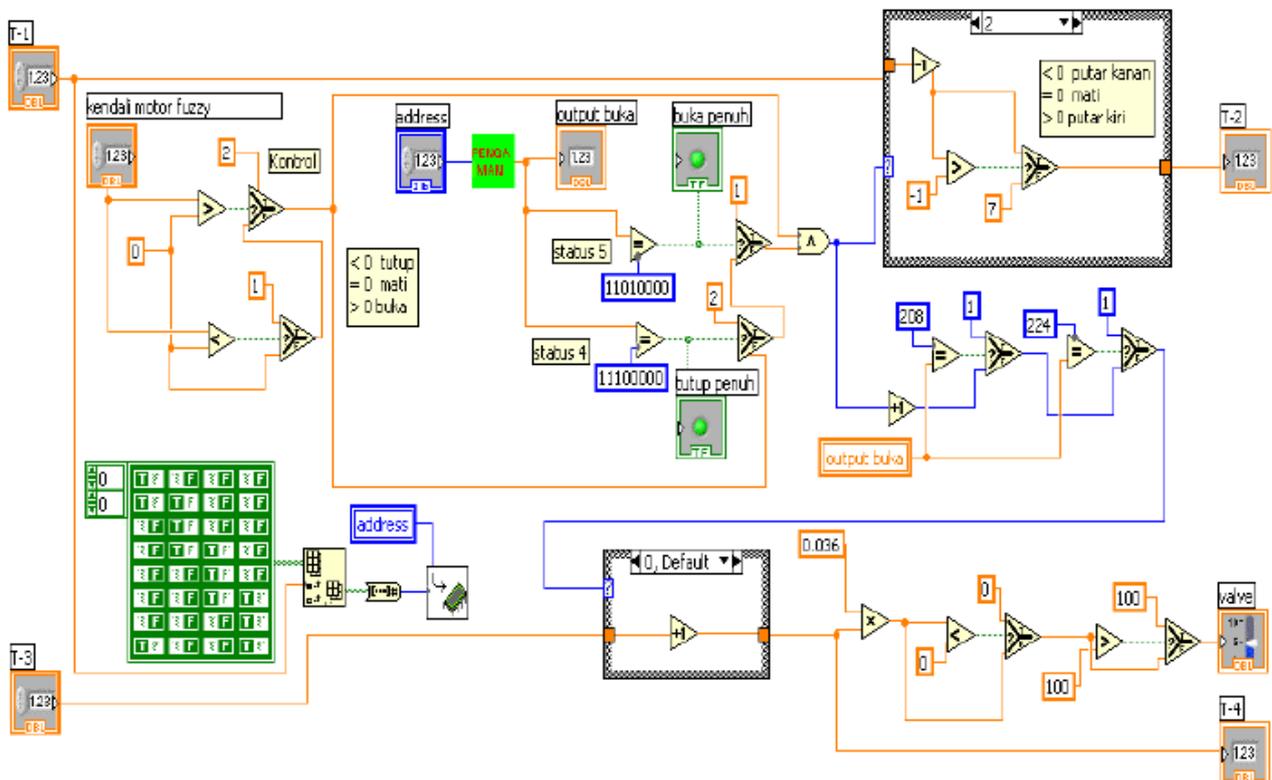
Gambar 15. Block diagram program tangka.vi



Gambar 16. Block diagram program pengaman.vi



Gambar 17. Block diagram program multiplexer.vi

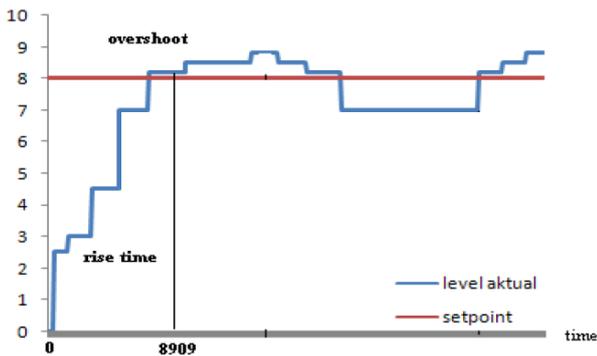


Gambar 18. Block diagram program motor stepper.vi

Tuning parameter FLC dilakukan dengan pertama-tama semut merancang jalur dari node awal ke node tujuan pada graph. Setelah satu jalur terselesaikan, maka hasil parameter tersebut akan diujikan pada sistem tangki proses. Nilai IAE dan ISE akan dicatat dan menjadi panduan update pheromone pada graph tersebut. Lalu proses pencarian jalur oleh semut akan diulang kembali beberapa iterasi hingga diperoleh hasil yang konvergen..

Pengujian terhadap performansi sistem dilakukan untuk mengetahui performansi sistem yang meliputi, lonjakan maksimum (*overshoot*) dan waktu naik (*rise time*). Pengujian terhadap performansi sistem dilakuakn dengan menentukan setpoint pada program pengontrol fuzzy.vi lalu dilakukan pengamatan pada level ketinggian air tangki proses, apakah dapat mencapai setpoint yang telah ditentukan serta mengamati performansi dari sistem apakah sesuai dengan kendali logika fuzzy yang dirancang, yaitu memperkecil lonjakan maksimum (*overshoot*) dan waktu naik (*rise time*).

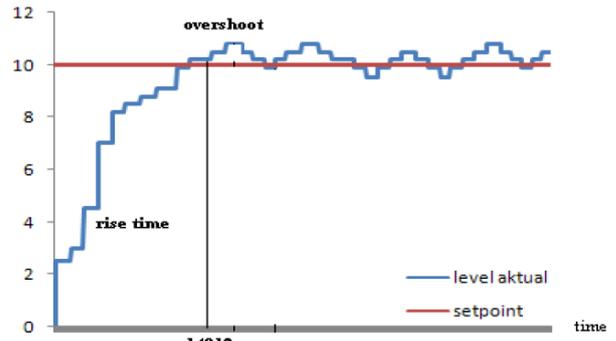
**Gambar 19** menunjukkan grafik performansi sistem dengan setpoint 8. **Gambar 20** menunjukkan grafik performansi sistem dengan setpoint 10. **Gambar 21** menunjukkan grafik performansi sistem dengan setpoint 15. **Gambar 22** menunjukkan grafik performansi sistem dengan memberikan setpoint yang berbeda, yaitu 8, 7 dan 9. **Table III** menunjukkan nilai *overshoot* dan *risetime* untuk setiap setpoint yang diuji cobakan.



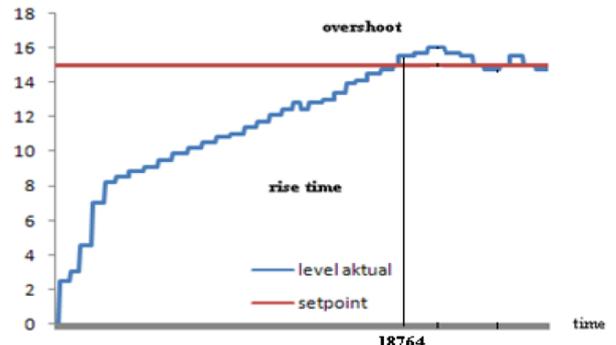
**Gambar 19.** Grafik performansi sistem dengan setpoint 8

**Table III.** Hasil pengujian performansi sistem

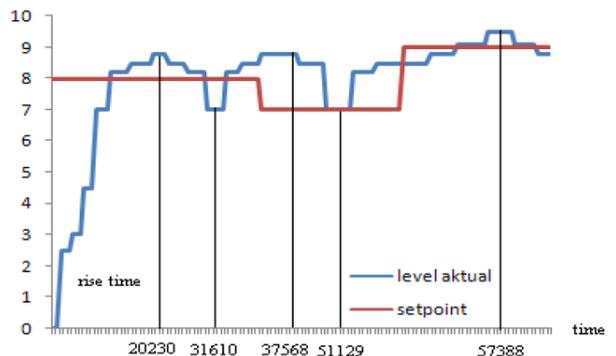
Setpoint	Overshoot	Risetime (ms)
8	2.5 %	8909
10	2 %	14812
15	3.33 %	18764



**Gambar 20.** Grafik performansi sistem dengan setpoint 10



**Gambar 21.** Grafik performansi sistem dengan setpoint 15



**Gambar 22.** Grafik performansi sistem dengan perubahan setpoint

Dari hasil pengujian performansi sistem dengan setpoint 8, 10 dan 15, dapat diamati bahwa sistem dapat berjalan sesuai dengan yang diharapkan, yaitu memiliki *overshoot* dan *risetime* yang kecil. *Overshoot* dapat diperkecil karena adanya kaidah 4 – 9 pada **Tabel I**. Adapun *risetime* dapat diperkecil karena adanya kaidah 1 – 3 pada **Tabel I**.

Pada pengujian sistem dengan memberikan perubahan pada setpoint, dapat diamati bahwa sistem merespon dengan baik, terlihat dari perubahan level aktual yang berubah sesuai dengan perubahan setpoint yang telah ditentukan.

### III. KESIMPULAN

Dari hasil pengujian, sistem pengontrol fuzzy dengan tuning parameter menggunakan ACO dapat berjalan dengan baik. Pengontrol fuzzy dapat mengatur putaran motor stepper sehingga

level ketinggian air yang diharapkan (*setpoint*) tercapai. Pengontrol fuzzy juga telah berhasil memperkecil lonjakan maksimum (*overshoot*) dan waktu naik (*rise time*). *Overshoot* dan *risetime* dapat diturunkan karena adanya kaidah fuzzy yang dipersiapkan untuk menurunkan *overshoot* dan *risetime*. Adapun penurunan *overshoot* dan *risetime* dapat dibuat minimal dikarenakan adanya proses tuning parameter fuzzy oleh algoritma ACO. Sebagai contoh, untuk *setpoint* 8, performansi keluaran sistem memiliki *overshoot* 2.5% dan *rise time* 8909 ms.

## DAFTAR PUSTAKA

- [1] Y. H. Chang, C. W. Chang, H. W. Lin, dan C. W. Tao, C. W., "Fuzzy controller design for ball and beam system with an improved ant colony optimization," *World Academy of Science, Engineering and Technology*, vol. 56, pp. 616-621, 2009
- [2] S. Tzeng, "GA approach for designing fuzzy control with nonlinear ball and beam system," *Proc. International Conference on Electrical Engineering (ICEE)*, vol. 5, pp. 1-8, 2008
- [3] J. Utama and S. Sitanggang. "Pengembangan Algoritma Pengolahan Citra Untuk Mengontrol Kepadatan Lalu Lintas Pada Persimpangan Dua Arah Berbasis Logika Fuzzy". Prosiding Digital SAINTIKS 2014," no. October 2014.
- [4] J. Brownlee, "Clever algorithms: nature-inspired programming recipes," *Creative Commons*, 2011
- [5] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, dan M. Valdez, "Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3196-3206, 2013
- [6] O. Castillo, H. Neyoy, J. Soria, M. García, dan F. Valdez, "Dynamic fuzzy logic parameter tuning for ACO and its application in the fuzzy logic control of an autonomous mobile robot," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, pp. 1-10, 2013
- [7] B. Yu, Z. Z. Yang, dan B. Yao, "An improved ant colony optimization for vehicle routing problem," *European journal of operational research*, vol. 196, no. 1, pp. 171-176, 2009
- [8] J. Brito, F. J. Martínez, J. A. Moreno, dan J. L. Verdegay, "An ACO hybrid metaheuristic for close-open vehicle routing problems with time windows and fuzzy constraints," *Applied Soft Computing*, vol. 32, pp. 154-163, 2015
- [9] L. N. Xing, Y. W. Chen, P. Wang, Q. S. Zhao, dan J. Xiong, "A knowledge-based ant colony optimization for flexible job shop scheduling problems," *Applied Soft Computing*, vol. 10, no. 3, pp. 888-896, 2010
- [10] W. Junqiang, dan O. Aijia, "A hybrid algorithm of ACO and delete-cross method for TSP," *2012 International Conference on Industrial Control and Electronics Engineering*, pp. 1694-1696, 2012
- [11] O. Castillo, H. Neyoy, J. Soria, P. Melin, dan F. Valdez, "A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot," *Applied soft computing*, vol. 28, pp. 150-159, 2015
- [12] R. Mitra, dan S. Singh, "Optimal fuzzy supervised PID controller using Ant colony optimization algorithm," *Advance in Electronic and Electric Engineering*, vol. 3, no. 5, pp. 553-560, 2013
- [13] J. Kaliannan, A. Baskaran, N. Dey, dan A. S. Ashour, "Ant colony optimization algorithm based PID controller for LFC of single area power system with non-linearity and boiler dynamics," *World Journal of Modelling and Simulation*, vol. 12, no. 1, pp. 3-14, 2016
- [14] D. Driankov, dan R. Palm, "Advances in fuzzy control", *Physica*, vol 16, 20103