

PENGEMBANGAN MODUL PEMBELAJARAN DE LORENZO PROCESS CONTROL DENGAN METODE JARINGAN SYARAF TIRUAN (JST) YANG BERBASIS OPEN SOURCE

Fahmi Abdillah Adji^{#1}, ArdikWijayanto ST. MT^{#2}, Ir. Wahjoe Tjatur S.^{#3}, MT, Ronny Susetyoko, S.Si, M.Si^{#4}

[#]Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya
Kampus PENS-ITS. Sukolilo, Surabaya

Abstrak--- Sistem kontrol memiliki tujuan untuk mengendalikan sesuatu sehingga sesuai atau mendekati dengan nilai parameter referensi yang diberikan. Seperti pada modul De Lorenzo yang telah menggunakan kontrol PID analog sebagai kontroler. Modul tersebut memiliki 4 sensor yaitu sensor level, sensor aliran air, sensor suhu dan sensor tekanan yang masing-masingnya dapat di atur oleh kontroler PID. Karena pentingnya kontrol cerdas dalam sistem yang tidak *linear* maka dalam Proyek Akhir (PA) ini ditambahkan kontroler yaitu dengan metode Jaringan Saraf Tiruan (JST). Metode JST ini yang akan dibandingkan dengan PID yang telah ada pada modul De Lorenzo. Parameter yang akan diatur adalah ketinggian air, kecepatan air dan suhu air. Dengan ini diharapkan nantinya mahasiswa dapat memahami tentang kontrol cerdas menggunakan JST dan membandingkan dengan PID. Dari hasil pengujian, PID konvensional dapat diandalkan untuk mendapatkan hasil yang maksimal. Tetapi PID bekerja maksimal jika mendapatkan Kp, Ki, dan Kd yang tepat. Metode JST backpropagation online, memiliki respon lebih lambat dibanding dengan PID dikarenakan JST melakukan pembelajaran bersamaan dengan proses kontrolnya (*online*). Dalam kontrol aliran air, PID dengan Kp, Ki, Kd tertentu dengan target aliran 60 L/h menghasilkan *rise time* = 2.4 detik dan *overshoot* = 1 L/h. Sedangkan dengan kontrol JST 1 lapis *hidden layer* dan 5 *neuron*, *rise time* lebih lambat yaitu 4.7 detik dengan *overshoot* = 1.5 L/h.

Kata kunci--- Kontrol Cerdas, JST, PID, Qt, Open Source

I. PENDAHULUAN

Laboratorium adalah sarana mahasiswa dalam belajar dan menerapkan teori yang telah dipelajari dalam perkuliahan. Praktikum tidak akan berjalan dengan lancar tanpa adanya modul belajar yang mendukung. Modul DL 2314 merupakan sarana mahasiswa untuk lebih mengenal tentang kontrol cerdas yang digunakan langsung pada sebuah miniatur plan, dimana pada modul tersebut telah terdapat kontroler PID yang dapat digunakan untuk kontrol level, kontrol aliran air, kontrol suhu, dan kontrol tekanan.

Suatu sistem kontrol diperlukan agar menjaga suatu besaran tertentu agar senantiasa sesuai dengan yang diinginkan. Untuk mengatur perilaku suatu sistem agar mengikuti kaidah yang diinginkan, serta untuk menjaga kestabilan sistem tersebut. Suatu sistem kontrol ternyata terdapat pada manusia untuk mengatur biologisnya. Kemampuan otak menyelesaikan suatu permasalahan, ternyata dipakai oleh suatu sistem kontrol yang disebut dengan Jaringan Saraf Tiruan(JST). JST adalah suatu model matematis yang tersusun dalam bentuk jaringan untuk

menyelesaikan permasalahan yang rumit dan tidak linear yang sulit untuk dimodelkan dalam bentuk matematis.

Dengan kelebihan JST dalam belajar ini, maka pada Proyek Akhir ini kami mencoba menggunakan JST sebagai sistem kontrol pada modul DL 2314, yaitu pada kontrol tinggi air, aliran air dan juga suhu. Dengan pengembangan modul de Lorenzo ini diharapkan nantinya mampu mempermudah mahasiswa dalam melakukan praktikum yang berhubungan dengan metode JST.

II. TINJAUAN PUSTAKA

A. DL 2314 Process Control

Modul De Lorenzo DL 2314 memiliki 4 sensor. Sensor-sensor yang digunakan pada modul De Lorenzo DL 2314 adalah sebagai berikut:

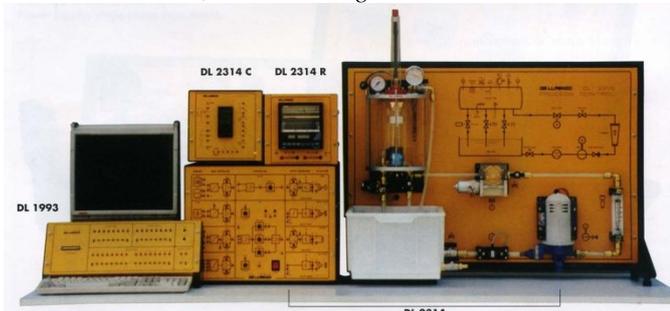
a. Temperature Sensor

- Platinum Thermo-resistance Pt 100
- Bi-metallic direct reading thermometer

b. Level Sensor

- Linear variable – differential transformer (LVDT)
- On-reed ON/OFF sensor

- c. *Flow Sensor*
 - *Flow meter 8000 pulses/liter*
 - *Flow meter, direct reading*
- d. *Pressure sensors*
 - *Strain gauge*
 - *Manometer, direct reading*



Gambar 1. De Lorenzo DL 2314

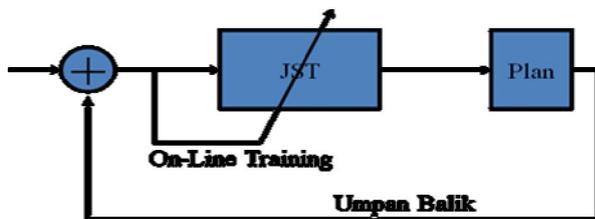
Selain sensor – sensor di atas, modul De Lorenzo ini memiliki:

- a. *Pressurized vessel capacity = 5 liters approx.*
- b. *Water tank capacity = 20 liters approx.*
- c. *Recirculation pump = 6 liters/minute, 12 V, 1.5 A.*
- d. *Motor driven valve = 4 manual valve*
- e. *Water heating resistance = 48V, 200W*
- f. *Safety valve set at 2.4 bar Safety thermostat Piping = brass*

Modul De Lorenzo telah dilengkapi oleh kontroler PID yang bisa digunakan sewaktu – waktu. Pada modul ini, kontrol PID digunakan untuk On – Off close loop kontrol level, close loop kontrol Flow, close loop kontrol On – Off temperatur.

B. JST

Jaringan Saraf Tiruan (JST) adalah suatu model matematis yang tersusun dalam bentuk jaringan untuk menyelesaikan permasalahan yang bersifat rumit dan tidak linear yang sulit dimodelkan dalam bentuk matematis. Prinsip kerjanya adalah kemampuan pembelajaran yang bersifat heuristik. Karena itu mudah terjebak dalam optimasi lokal.

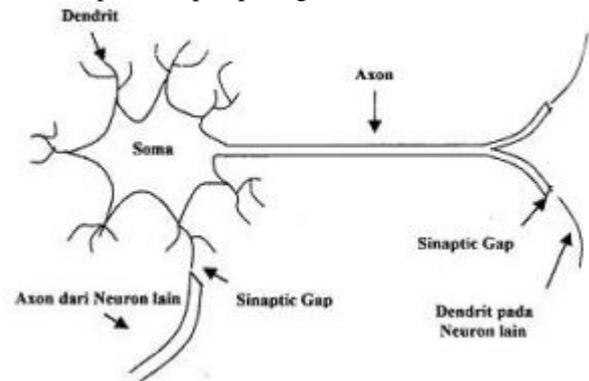


Gambar 2. JST Online Training

1) Model JST

JST bekerja seperti sistem syaraf biologis pada saat berhubungan dengan dunia luar. Sel syaraf biologis atau disebut juga Neuron memiliki tiga komponen penyusun yang saling bekerja sama untuk mengolah sinyal-sinyal informasi.

Tiga komponen tersebut adalah *Dendrit*, *soma* atau badan sel dan *axon* seperti tampak pada gambar 3.



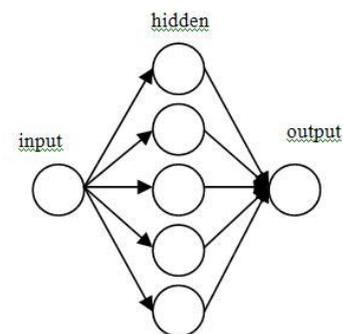
Gambar 3. Struktur Sel Syaraf biologis

Untuk mengirimkan informasi ke sel lain, sinyal dilewatkan melalui *axon* yang merupakan serabut syaraf tunggal dan pada ujungnya bercabang-cabang. Selanjutnya sinyal akan melalui celah sinapsis dan disampaikan ke soma lain oleh *dendrit neuron* tersebut. Sehingga dengan integrasi dan terkoneksiya *neuron* satu dengan yang lain akan dapat melaksanakan aktifitas secara teratur sesuai kebutuhan.

JST memiliki arsitektur dan algoritma pembelajaran yang beragam (cukup banyak), salah satu contohnya adalah BNN(Back-propagation Neural Network).

2) Jenis –jenis JST

Jaringan Syaraf Tiruan (JST) bertujuan untuk meniru kemampuan manusia dalam beradaptasi dengan lingkungannya dan belajar dari pengalaman. Jaringan ini terdiri dari banyak simpul (*node*) pemroses yang dianalogikan dengan *neuron* pada otak. Proses matematis yang ada dalam jaringan ini juga merupakan pendekatan pada cara kerja otak. Inti dari proses pembelajaran pada Jaringan Syaraf Tiruan adalah untuk meminimumkan error antara keluaran sistem dengan keluaran JST.



Gambar 4. Neuron dalam JST

Dalam proses kerja JST, kemampuan kerjanya paling dipengaruhi oleh proses belajarnya. JST harus menyesuaikan bobot-bobot untuk melakukan pengolahan terhadap sinyal-sinyal masukannya. Kemampuan untuk menyesuaikan bobot-bobot itulah yang menentukan kemampuannya dalam belajar. Penyesuaian harga bobot-bobot itu biasa dilakukan dengan

cara memperbaharui dalam setiap iterasi sampai bobot-bobot yang paling sesuai dicapai.

Jenis dari JST dapat dibedakan berdasarkan apakah JST belajar sendiri atau dipandu.

- **Supervised**

Proses pembelajaran JST diarahkan sesuai dengan keinginan (memiliki hasil tertentu). JST model ini banyak digunakan untuk pengenalan, sistem kendali atau segala sesuatu yang memerlukan kepastian hasil.

- **Unsupervised**

Proses pembelajaran JST dilakukan dengan sendirinya (memiliki hasil tertentu). JST model ini banyak digunakan pada sistem statistik atau stokastik, misalnya pengelompokan (Clustering).

Terdapat berbagai proses pembelajaran yang dapat diterapkan pada Jaringan Saraf Tiruan, seperti algoritma pembelajaran Backpropagation atau propagasi balik.

3) Neuron dalam JST

Neuron memiliki beberapa input yang berasal dari unit sensor atau neuron lain. Inti neuron akan menjumlahkan sinyal – sinyal masuk tersebut. Untuk menjadikan setiap neuron sebagai sel pengolah yang bersifat non linear, serta mencegah adanya tingkat energi yang berlebihan, maka pada keluaran neuron ditambahkan suatu fungsi aktivasi tertentu.

Inti Neuron memiliki beberapa input, dimana setiap input memiliki karakteristik transfer yang berbeda yang dinyatakan dalam bentuk penguatan dan dikenal dengan istilah bobot (weight). Untuk mengubah tingkat energi dalam suatu neuron digunakan suatu bias. Setiap neuron memiliki satu output yang akan tersambung dengan neuron lain, tetapi dengan tingkat energi yang dibatasi (menggunakan fungsi aktivasi).

- **Lapisan Input**

Setiap input sensor akan tersambung dengan lapisan neuron bagian depan. Lapisan neuron bagian depan ini dapat dianggap sebagai lapisan input dan biasanya terdiri dari sejumlah neuron yang jumlahnya sama dengan input sensor. Lapisan ini berfungsi untuk menampung dan menyebarkan sinyal yang ada ke lapisan berikutnya.

Lapisan Tersembunyi

Setiap neuron lapis input tersambung dengan sambungan lengkap dengan sejumlah lapis neuron bagian dalam. Lapis ini dianggap sebagai lapis tengah atau lapis tersembunyi. Lapisan ini berada di antara lapisan input dan lapisan output. Lapis tersembunyi ini dapat terdiri dari satu atau beberapa lapis tersembunyi, dimana setiap lapis dapat terdiri dari sejumlah neuron. Jumlah dari lapisan tersembunyi menentukan derajat fungsi pemetaan yang dapat dilakukan oleh JST. Jumlah neuron tiap lapis menentukan seberapa banyak permasalahan (data masukan) yang dapat diinginkan. Namun semakin banyak jumlah lapis tersembunyi ini atau semakin banyak jumlah neuron tiap lapis akan meningkatkan kompleksitas sistem JST itu sendiri dan dapat menurunkan kecepatan pembelajaran JST.

- **Lapisan Output**

Lapisan neuron bagian akhir akan tersambung secara langsung pada bagian keluaran (tahap output) dan disebut sebagai lapis output. Jumlah neuron pada lapis output tergantung dari jumlah output yang diinginkan. Neuron pada bagian lapis akhir dapat berbentuk neuron biasa, atau hanya sekedar untuk mengkodekan dari keluaran JST ke bentuk keluaran yang diinginkan.

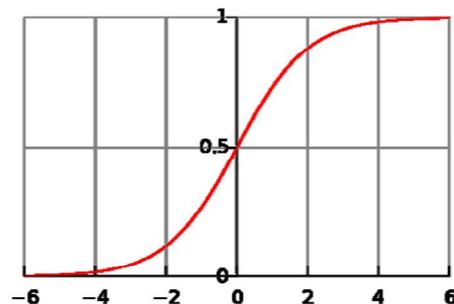
4) Fungsi Aktivasi

Fungsi aktivasi mentransformasi nilai keluarannya melalui pemetaan sinyal masukannya ke dalam sebuah nilai yang sesuai dengan neuron lainnya.

- **Sigmoid**

Fungsi aktivasi yang dipakai harus memenuhi beberapa syarat, yaitu: kontinu, dapat didiferensiasi, dan secara monoton tidak menurun. Salah satu fungsi yang sesuai adalah fungsi Sigmoid. Fungsi sigmoid ada 2, fungsi sigmoid biner dengan range (0,1)

Fungsi aktivasi sigmoid biner dinyatakan dengan persamaan sebagai berikut:
$$P(t) = \frac{1}{1+e^{-t}}$$



Gambar 4. Sigmoid biner

5) Back-propagation

Algoritma backpropagasi atau disebut sebagai propagasi balik, pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada JST, dan selanjutnya algoritma ini biasa disingkat dengan BP. Algoritma ini termasuk metode pelatihan supervised.

Algoritma ini juga banyak dipakai pada aplikasi pengaturan karena proses pelatihannya didasarkan pada hubungan yang sederhana, yaitu jika keluaran memberikan hasil yang salah dari target, maka penimbang atau bobot dan bias dikoreksi supaya nilai kesalahan selanjutnya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang diinginkan.

Secara garis besar, backpropagation ini dapat dideskripsikan sebagai berikut: ketika jaringan diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit lapisan keluaran. Kemudian unit-unit lapisan keluaran memberikan tanggapan yang disebut sebagai keluaran jaringan. Saat keluaran jaringan tidak sama dengan keluaran yang diharapkan maka keluaran akan melakukan perhitungan mundur (backward) pada lapisan tersembunyi diteruskan ke unit pada lapisan masukan. Oleh karenanya maka algoritma

pelatihan ini dinamakan propagasi balik atau backpropagation.

Algoritma pelatihan backpropagation terdiri dari dua proses, propagasi maju dan propagasi mundur dari error.

1. Inisialisasi bobot secara acak.
2. Propagasi maju:
 - a. Masing-masing unit masukan menerima sinyal masukan dan sinyal tersebut disebarkan ke unit-unit lapisan tersembunyi.
 - b. Masing-masing unit lapisan tersembunyi, dikalikan dengan bobotnya dan dijumlahkan serta ditambah dengan biasnya.

$$Z_{in_j} = V_{o_j} + \sum_{i=1}^m X_i V_{ij}$$

- c. Untuk mengeluarkan sinyal, unit-unit tersebut harus diaktifkan dengan fungsi aktivasi:

$$Z_j = f(Z_{in_j})$$

Jika fungsi aktivasi yang digunakan adalah fungsi sigmoid maka bentuk fungsi tersebut adalah:

$$Z_j = \frac{1}{1 + e^{-Z_{in_j}}}$$

- d. Kemudian sinyal-sinyal tersebut dikirimkan ke unit lapisan keluaran.
- e. Masing-masing unit keluaran dikalikan dengan bobotnya dan dijumlahkan serta ditambah dengan biasnya

$$Y_{in_k} = W_{o_k} + \sum_{j=1}^p Z_j W_{jk}$$

- f. Setiap unit keluaran diaktifkan dengan fungsi aktivasi sehingga menghasilkan output JST.

$$y_k = f(Y_{in_k})$$

3. Propagasi mundur
 - a. Menghitung error
 - b. Masing-masing unit keluaran ($y_k, k=1, \dots, m$) menerima pola target sesuai dengan pola masukan saat pelatihan dan dihitung errornya

$$\delta_k = (t_k - y_k) f'(Y_{in_k})$$

Karena $f'(Y_{in_k})$ menggunakan fungsi sigmoid, maka:

$$\begin{aligned} f'(Y_{in_k}) &= f'(Y_{in_k})(1 - f'(Y_{in_k})) \\ &= y_k(1 - y_k) \end{aligned}$$

- c. Menghitung perbaikan bobot:

$$\Delta W_{kj} = \alpha \cdot \delta_k \cdot Z_j$$
- d. Menghitung perbaikan bias:

$$\Delta W_{o_k} = \alpha \cdot \delta_k$$
- e. Dan menggunakan δ_k pada semua unit lapisan sebelumnya.
- f. Masing-masing penimbang yang menghubungkan unit-unit lapisan keluaran dengan unit-unit pada lapisan tersembunyi ($Z_j,$

$j=1, \dots, p$) dikalikan dengan delta dan dijumlahkan sebagai masukan ke unit-unit lapisan berikutnya.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk}$$

- g. Selanjutnya dikalikan dengan turunan dari fungsi aktifasinya untuk menghitung error.

$$\delta_j = \delta_{in_j} f'(Y_{in_j})$$

- h. Kemudian menghitung perbaikan bobotnya

$$\Delta W_{ij} = \alpha \cdot \delta_j \cdot X_i$$

- i. Kemudian menghitung perbaikan biasnya

$$\Delta V_{o_j} = \alpha \cdot \delta_j$$

- j. Masing-masing keluaran unit ($y_k, k=1, \dots, m$) diperbaiki bias dan bobotnya ($j=0, \dots, p$),

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk}$$

- k. Masing-masing unit tersembunyi ($Z_j, j=1, \dots, p$) diperbaiki bias dan bobotnya ($j=0, \dots, n$).

$$V_{jk}(\text{baru}) = V_{jk}(\text{lama}) + \Delta V_{jk}$$

Keterangan:

α = learning rate $0 < \alpha < 1$

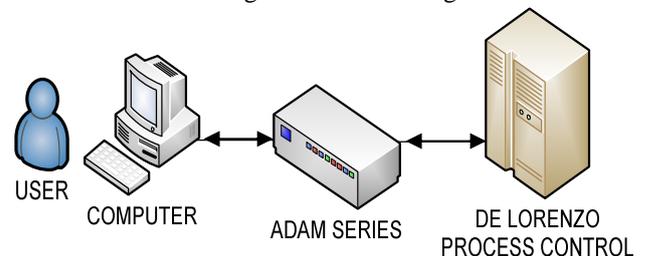
δ_j = faktor pengaturan nilai bobot sambungan pada lapisan tersembunyi.

δ_k = faktor pengaturan nilai bobot sambungan pada lapisan keluaran.

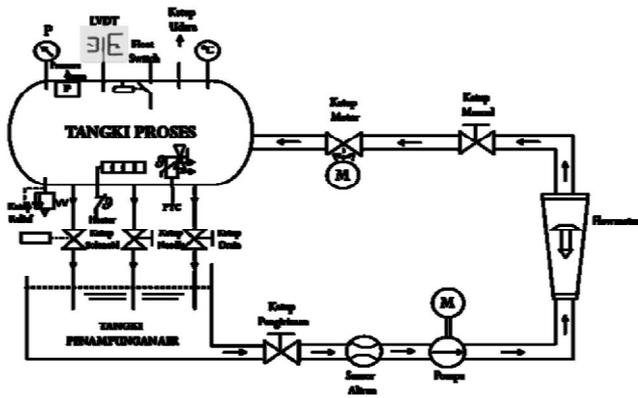
III. PERENCANAAN SISTEM

Sistem terbentuk dari Modul De Lorenzo DL 2314 Process Control sebagai plan nya dan Modul ADAM 5000/485 sebagai akuisisi data dan aksi. ADAM terhubung langsung dengan modul De Lorenzo. Data yang diterima ADAM dikirim ke komputer dengan komunikasi serial. Pada komputer telah ada GUI yang menampilkan data analog input dan beberapa setting analog output. Data analog tersebut nantinya diolah sehingga dapat ditampilkan pada layar GUI. Dalam penggunaan kontrol proses ini, input *set point* akan dimasukkan dalam program. Nilai keluaran akan dibandingkan dengan *set point* yang diberikan lalu diproses sampai nantinya akan menghasilkan nilai keluaran yang mendekati *set point*.

Berikut ini adalah gambaran blok diagram sistem:

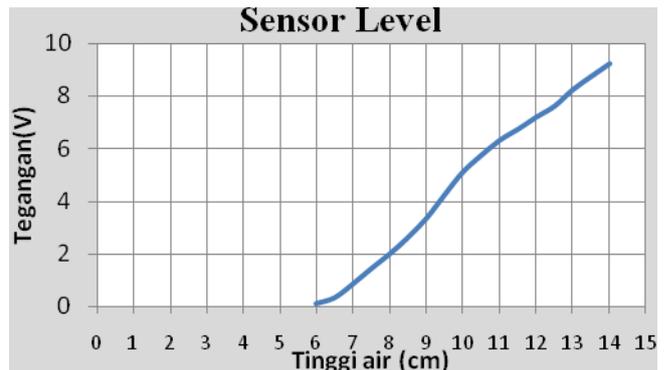


Gambar 6. Diagram Blok Sistem



Gambar 7. Blok Diagram Plan De Lorenzo DL 2341

• **Sensor Level**



Gambar 3. Karakteristik Level Sensor

Dari data yang telah diambil, maka didapatkan persamaan untuk level sensor sebagai berikut:

$$y = -7.48 + 1.22 x$$

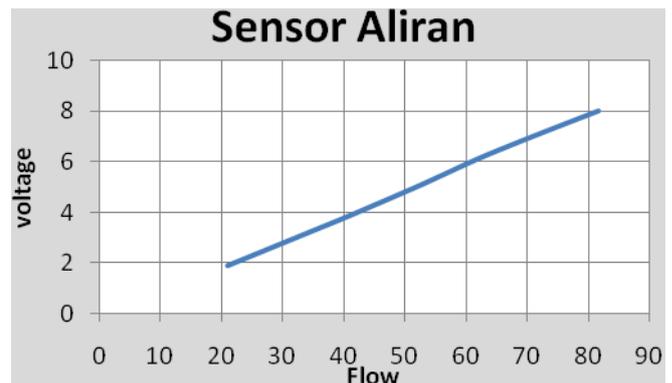
Keterangan:

y = Tegangan (V)

x = Tinggi air (cm)

Setiap kenaikan tinggi air sebesar 1cm, maka sensor ada kecenderungan naik sebesar 1.22 volt.

• **Sensor Flow**



Gambar 4. Karakteristik Sensor Flow

Didapatkan persamaan dari sensor flow adalah sebagai berikut:

$$y = -0.253 + 0.102x$$

Keterangan:

y = tegangan (sensor flow)

x = aliran air (l/h)

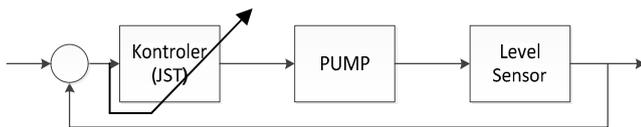
Jadi, sensor memiliki kecenderungan naik sebesar 0.102 v setiap kenaikan aliran air sebesar 1 l/h.

• **Sensor Temperatur**

Pembacaan temperature menggunakan thermometer yang telah ada pada plan, dan sensor PTC. Berikut ini adalah table temperature.

A. **Kendali Level dengan Metode JST**

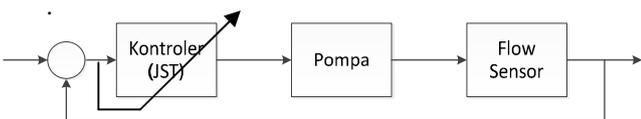
Pengendalian level pada tank dilakukan dengan mengatur kecepatan pada pompa. Untuk menaikkan level air, tangki proses akan disuplai oleh tangki penampungan dengan menyalakan pompa. Ketinggian level akan dibaca oleh sensor level berupa LVDT. Turun naiknya level air akan dideteksi oleh sensor dan akan dirubah menjadi besaran tegangan.



Gambar 8. Blok diagram kendali level dengan JST sebagai kontroler

B. **Kendali aliran dengan metode JST**

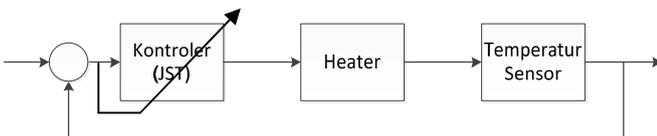
Pengendalian kecepatan aliran air dilakukan dengan mengatur kecepatan dari pompa. Perubahan kecepatan pompa menyebabkan perubahan kecepatan aliran air. Perubahan dari kecepatan aliran akan dibaca oleh sensor IR Opflow dan dirubah dalam besaran tegangan.



Gambar 9. Blok diagram kendali aliran dengan JST sebagai kontroler

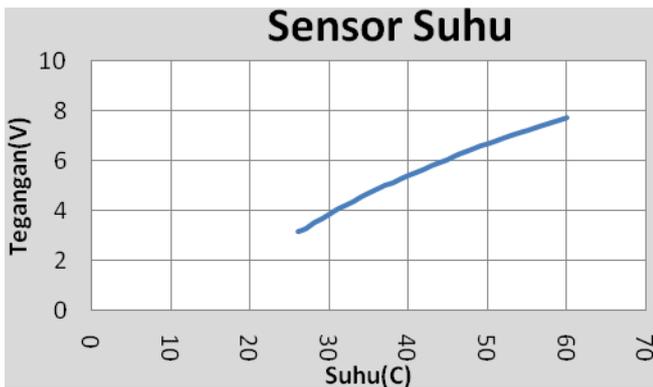
C. **Kendali suhu dengan metode JST**

Pengendalian suhu air dilakukan dengan mengatur heater sebagai pemanasnya.



Gambar 2. Blok diagram kendali suhu dengan JST sebagai kontroler

D. **Observasi Modul DL 2314:**



Gambar 5. Karakteristik Sensor Temperatur

Dari data tersebut didapatkan persamaan sensor suhu sebagai berikut:

$$y = -1.31 + 0.171x$$

Keterangan:

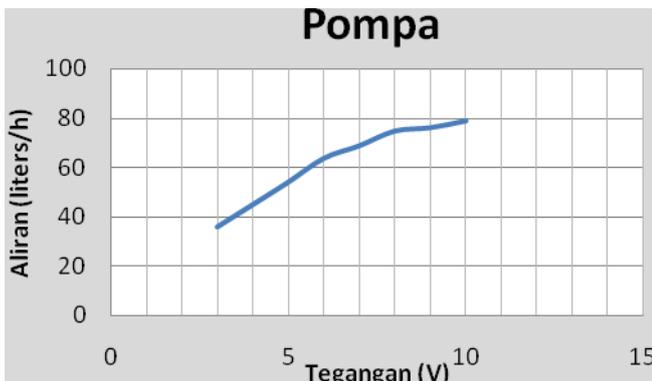
y = tegangan (V – sensor suhu)

x = suhu (°C)

Sensor akan cenderung naik 0.171 v setiap kenaikan suhu sebesar 1°C.

• Karakteristik pompa

Pompa akan menghasilkan kecepatan aliran yang berbeda setiap tegangan yang diberikan kepada pompa tersebut. Karakteristik pompa dapat dilihat pada tabel hasil observasi berikut ini:



Gambar 64. Karakteristik Pompa

IV. HASIL DAN ANALISA

A. Pengujian Software

1) Kontrol level menggunakan JST Backpropagasi Online

Dalam proyek akhir ini, kontrol JST akan digunakan dalam 3 hal, yaitu kontrol level air, kontrol aliran air, dan kontrol suhu air yang akan dilakukan secara SISO. Pengujian pertama adalah kontrol JST backpropagasi secara online untuk mengontrol ketinggian air yang berada pada tangki proses. Waktu cuplik yang digunakan adalah 200 milisecond. Jadi dalam pengambilan datanya, dilakukan setiap 200 ms. Dalam kontrol level ini, tegangan yang diberikan pada pompa berkisar 0-5 volt, dengan aliran 0 L/h sampai dengan 100 L/h.

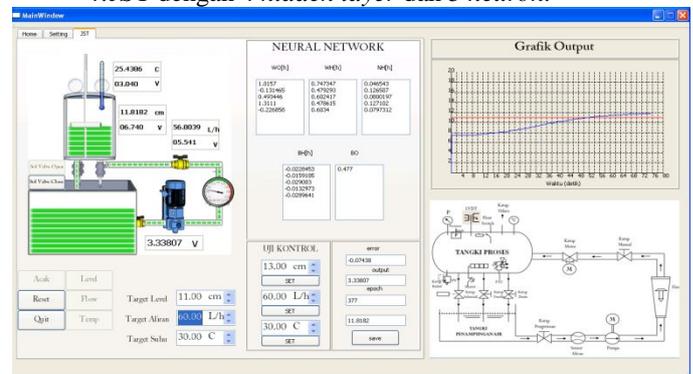
Tujuan dari pengujian ini adalah untuk mengetahui bagaimana respon JST dalam kontrol level air. Pada pengujian ini, juga dilakukan pengujian bagaimana pengaruh jumlah hidden layer dan learning rate terhadap respon kontrol JST.

• Pengaruh jumlah hidden layer terhadap respon sistem

Dalam beberapa referensi, penentuan jumlah hidden layer yang digunakan dalam menyelesaikan suatu permasalahan tidak dijelaskan secara jelas. Dalam hal ini tidak ada panduan yang menjelaskan mengenai berapa hidden layer yang dibutuhkan untuk menjadikan JST menjadi lebih baik.

Pada proyek akhir JST akan diuji dengan jumlah hidden layer yang berbeda-beda. Berikut ini adalah beberapa macam kondisi setting pada JST yang akan di uji cobakan pada kontrol level ini, yaitu:

1. JST dengan 1 hidden layer dan 5 neuron
2. JST dengan 2 hidden layer dan 5 neuron
3. JST dengan 3 hidden layer dan 5 neuron
4. JST dengan 4 hidden layer dan 5 neuron.

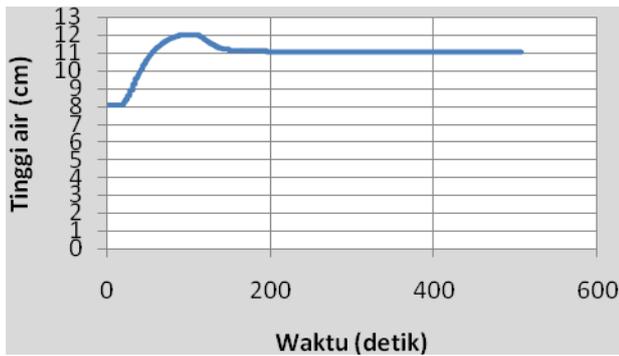


Gambar 15. GUI dalam proses kontrol level.

Gambar di atas merupakan tampilan GUI yang digunakan dalam proses kontrol level air menggunakan metode jaringan saraf tiruan backpropagasi.

Untuk pengujian yang pertama adalah dengan menggunakan JST dengan kondisi sebagai berikut:

- Hidden layer = 1
- Neuron = 5
- Learning Rate = 0.1
- Input layer = 1
- Output layer = 1
- Target = 11 cm



Gambar 16. Hasil pengujian pertama kontrol level dengan kondisi 1.

Pengujian dengan kondisi pertama di lakukan sebanyak 5 kali. Gambar di atas merupakan 1 hasil pengujian dari 5 pengujian yang telah dilakukan. Untuk lebih jelasnya, dapat dilihat pada tabel 1.

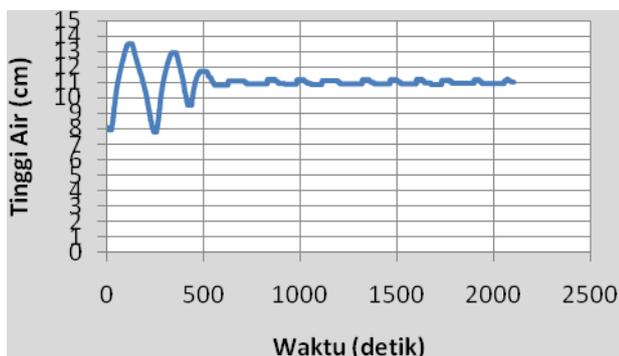
Tabel 1. Hasil pengujian kontrol level dengan JST kondisi 1.

No	Set Point (cm)	Rise time (detik)	Overshoot (cm)	Settling Time (detik)
1	11	27.8	0.972	135.6
2	11	33	1.186	165.6
3	11	32	1.001	156
4	11	28.8	0.975	112.4
5	11	27	2.0643	150.4

5 pengujian yang telah dilakukan ini, semuanya memiliki respon yang serupa. Respon yang dihasilkan memiliki overshoot yang tidak terlalu besar dan rise time yang lama. JST kondisi 1 ini mampu mencapai target dalam waktu antara 112.4 – 165.6 detik.

Pengujian kedua yaitu dengan kondisi JST memiliki 2 hidden layer dan 5 neuron setiap hidden layer nya. Seperti pada pengujian sebelumnya, target yang diberikan adalah 11 cm.

- Hidden layer = 2
- Neuron = 5
- Learning Rate = 0.1
- Input layer = 1
- Output layer = 1
- Target = 11 cm



Gambar 17. Hasil pengujian kedua kontrol level dengan kondisi 2.

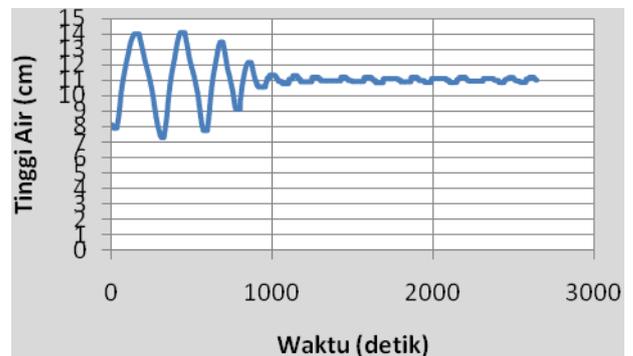
Tabel 2. Hasil pengujian kontrol level dengan JST kondisi 2.

No	Set Point (cm)	Rise time (detik)	Overshoot (cm)	Settling Time (detik)
1	11	24	2.768	730
2	11	24.2	3.038	1031.6
3	11	23.2	3.668	3102.6
4	11	25.2	3.537	1187
5	11	24.2	2.512	510

Pengujian kedua menghasilkan respon yang memiliki rise time yang lebih cepat dari pada pengujian yang pertama. Tetapi dengan rise time yang cepat ini, JST membutuhkan waktu yang lebih lama untuk mencapai target. Dapat dilihat dari settling time yang dihasilkan yaitu berkisar 730 – 3102 detik. Jauh berbeda dengan JST kondisi 1.

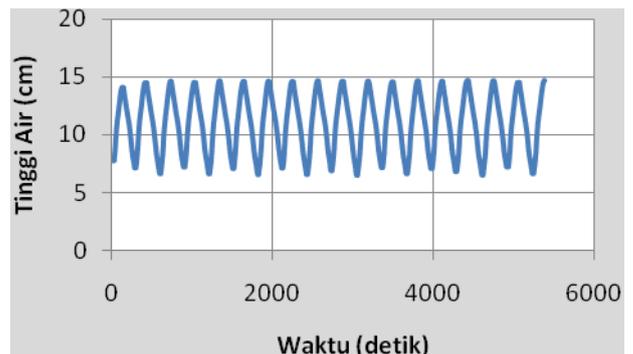
Pengujian ketiga dilakukan dengan kondisi JST sebagai berikut:

- Hidden layer = 3
- Neuron = 5
- Learning Rate = 0.1
- Input layer = 1
- Output layer = 1
- Target = 11 cm



Gambar 18. Hasil pengujian pertama kontrol level dengan kondisi 3.

Pengujian ketiga menghasilkan rise time yang sedikit lebih lama dari pada respon pengujian kedua. Dari 4 pengujian yang dilakukan pada kondisi ketiga ini, ada saat dimana JST tidak dapat mencapai kondisi stabil atau mengalami isolasi secara terus menerus.



Gambar 19. Hasil pengujian ketiga kontrol level dengan kondisi 3.

Tabel 3. Hasil pengujian kontrol level dengan JST kondisi 3.

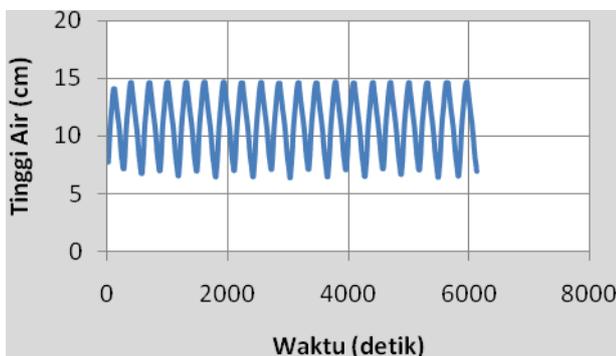
No	Set Point (cm)	Rise time (detik)	Overshoot (cm)	Settling Time (detik)
1	11	27	3.062	1087.2

2	11	26	3.27	895.6
3	11	-	-	-
4	11	25.6	2.472	478.8

Pengujian keempat dilakukan dengan kondisi JST sebagai berikut:

- *Hidden layer* = 4
- *Neuron* = 5
- *Learning Rate* = 0.1
- *Input layer* = 1
- *Output layer* = 1
- *Target* = 11 cm

Mengacu pada pengujian yang sebelumnya, dimana JST pernah mengalami keadaan osilasi terus menerus. Di pengujian keempat ini dengan semakin banyaknya *hidden layer* yang digunakan seharusnya hal yang pernah terjadi pada kondisi 3 akan terjadi pada kondisi 4. Berikut ini adalah hasil pengujian yang telah dilakukan.



Gambar 20. Hasil pengujian pertama kontrol level dengan kondisi 4.

Tabel 4. Hasil pengujian kontrol level dengan JST kondisi 4.

No	Set Point (cm)	Rise time (detik)	Overshoot (cm)	Settling Time (detik)
1	11	-	-	-
2	11	-	-	-
3	11	-	-	-
4	11	-	-	-

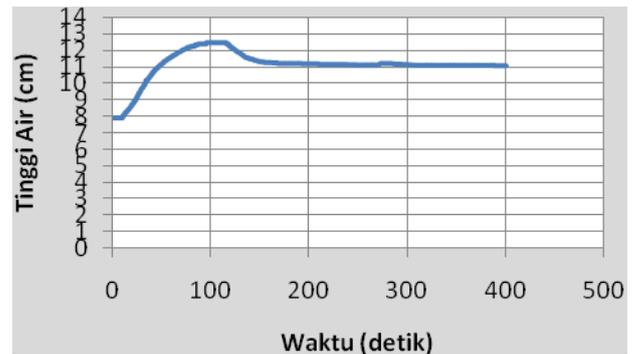
Dari 4 pengujian yang telah dilakukan, semuanya gagal untuk mengontrol level air. Semakin banyaknya *hidden layer* yang digunakan, akan memperlambat kinerja dari JST tersebut bahkan membuat JST tidak dapat mengontrol sistem.. Hal ini terjadi pada kontrol JST dengan kondisi keempat (4 *hidden layer*).

Banyak hal yang mempengaruhi respon dari kontrol JST pada sebuah plan. Banyaknya jumlah neuron pada hidden layer dan ketebalannya, mempengaruhi bagaimana JST tersebut bekerja. Semakin banyak jumlah neuron pada hidden layer tersebut, semakin kompleks JST tersebut bekerja tetapi juga dapat memperlambat kinerja dari JST. Selain itu, kinerja JST juga dipengaruhi dengan nilai acak bobot dan bias pada awal inialisasi.

- **Pengaruh learning rate terhadap respon sistem**

Kinerja JST juga dipengaruhi oleh *learning rate* yang digunakan. Untuk mengetahui pengaruh nilai *learning rate* pada kinerja JST dilakukan dengan cara mengubah nilai *learning rate* yang berbeda, sementara parameter lainnya diberikan dengan nilai tetap. Berikut ini adalah respon kontrol level JST dengan *learning rate* yang berbeda.

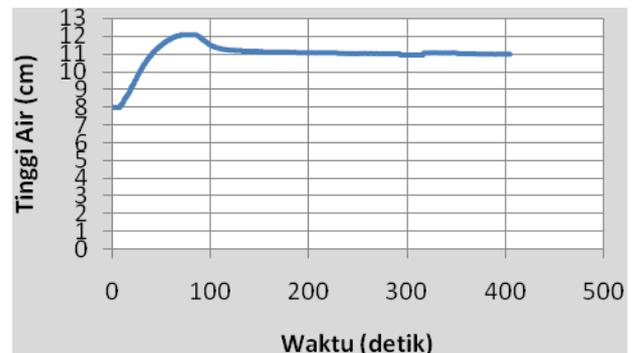
- *Hidden layer* = 1
- *Neuron* = 5
- *Learning rate* = 0.08
- *Output layer* = 1
- *Input layer* = 1
- *Target* = 11 cm



Gambar 21. Hasil pengujian JST dengan *learning rate* 0.08

Nilai *learning rate* dinaikkan menjadi 0.1, dan nilai acak bobot dan bias awal disamakan dengan sebelumnya. Kondisi JST adalah sebagai berikut:

- *Hidden layer* = 1
- *Neuron* = 5
- *Learning rate* = 0.2
- *Output layer* = 1
- *Input layer* = 1
- *Target* = 11 cm



Gambar 22. Hasil pengujian JST dengan *learning rate* 0.2

Tabel 5. Pengaruh learning rate terhadap respon sistem.

No	Learning rate	Set Point (cm)	Rise time (detik)	Overshoot (cm)	Settling Time (detik)
1	0.08	11	43.4	12.434	167.8
2	0.1	11	39.4	12.285	151.2
3	0.2	11	39.2	12.079	124.8

4	0.4	11	38.8	11.91	108.4
---	-----	----	------	-------	-------

Berdasarkan tabel 5 di atas dan gambar grafik respon sistem, penambahan nilai *learning rate* akan berpengaruh terhadap kecepatan respon sistem. Semakin besar nilai *learning rate* nya semakin cepat plan mencapai target.

Dari tabel Dengan hasil ini maka dapat diketahui bahwa pengaruh banyaknya *hidden layer* dan *learning rate* yang digunakan terhadap respon kontrol pada kontrol level atau ketinggian air adalah sebagai berikut:

- Semakin banyak *hidden layer* yang digunakan, maka semakin lambat proses belajar dalam kontrol level air.
- Semakin banyak *hidden layer* yang digunakan, belum tentu dapat mengontrol lebih baik. Bisa jadi kontroler (JST) terjebak dalam proses pembelajarannya atau terus beresilasi.
- Semakin besar nilai dari *learning rate* respon sistem untuk mencapai target semakin cepat.

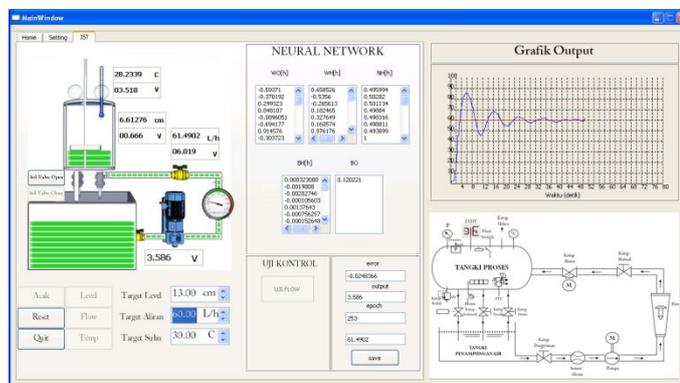
2) Kontrol aliran air menggunakan JST

Pengujian selanjutnya adalah menguji perangkat lunak Jaringan Saraf Tiruan sebagai kontrol aliran air. Waktu cuplik yang digunakan adalah 200 ms. Aktuator pada pengujian ini adalah pompa yang diberikan tegangan berkisar 0-5 volt, dengan aliran berkisar 0 L/h sampai dengan 100 L/h.

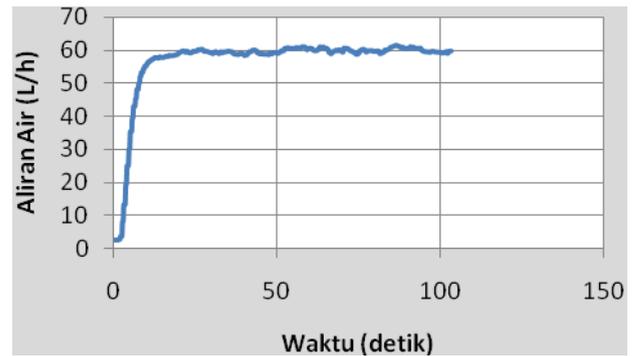
• Pengaruh jumlah hidden layer terhadap respon sistem

Pengujian dilakukan dengan kondisi yang berbeda-beda. Seperti pada halnya kontrol level sebelumnya. Pada kontrol aliran air ini, dilakukan beberapa kali percobaan dengan jumlah lapis *hidden layer* dan neuron pada *hidden layer* yang beragam. Disini akan diketahui apakah keadaan osilasi dikarenakan semakin banyaknya *hidden layer* yang terjadi pada kontrol level akan terjadi pada kontrol aliran air ini atau tidak. Kodisi JST yang akan di uji adalah sebagai berikut:

- JST dengan 1 *hidden layer* dan 5 neuron
- JST dengan 2 *hidden layer* dan 5 neuron
- JST dengan 3 *hidden layer* dan 5 neuron
- JST dengan 4 *hidden layer* dan 5 neuron
- JST dengan 1 *hidden layer* dan 10 neuron
- JST dengan 2 *hidden layer* dan 10 neuron
- JST dengan 3 *hidden layer* dan 10 neuron



Gambar 23. GUI dalam proses kontrol aliran



Gambar 24. Hasil pengujian pertama kontrol aliran dengan kondisi 1.

Pengujian di atas adalah 1 dari 10 pengujian kontrol aliran dengan JST kondisi 1 yang telah dilakukan. Data-data hasil pengujian pada kondisi 1 ini dapat dilihat pada tabel 6 di bawah ini.

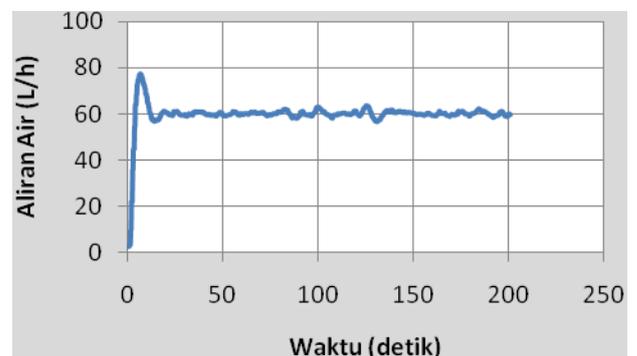
Tabel 6. Hasil pengujian kontrol aliran dengan JST kondisi 1.

No	Set Point (L/h)	Rise time (detik)	Overshoot (L/h)
1	60	6.4	1,519
2	60	2	12,902
3	60	4.4	1.598
4	60	8.8	3.167
5	60	4.6	1.98
6	60	6.4	3.088
7	60	4.1	2.49
8	60	7.4	2.47
9	60	4.4	3.823
10	60	4.7	1.52

Respon dari pengujian kontrol aliran pada kondisi pertama memiliki *overshoot* yang kecil dan *rise time* yang lama. Tetapi hal ini membuat JST mampu lebih cepat mencapai *steady state*.

Pengujian kondisi kedua:

- *Hidden layer* = 2
- *Neuron* = 5
- *Learning rate* = 0.1
- *Input layer* = 1
- *Output layer* = 1
- *Target* = 60 L/h

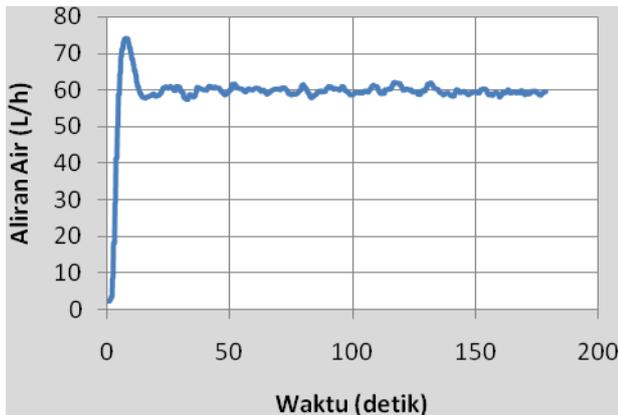


Gambar 25. Hasil pengujian ketiga kontrol aliran dengan kondisi 2

Tabel 7. Hasil pengujian kontrol aliran dengan JST kondisi 2

No	Set Point (L/h)	Rise time (detik)	Overshoot (L/h)
1	60	9	2.059
2	60	3.4	7.314
3	60	2	17.206
4	60	3.8	4.578
5	60	1.8	19.157
6	60	8.6	3.088
7	60	6	3.284
8	60	1.8	20.57
9	60	7.4	3.392
10	60	4	8.833

Dibandingkan dengan kondisi pertama, kondisi kedua ini lebih sering menghasilkan *overshoot* yang besar dan *rise time* yang lebih cepat daripada kondisi pertama.

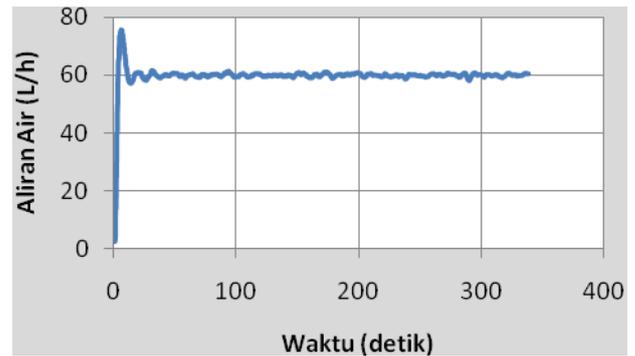


Gambar 26 Hasil pengujian pertama kontrol aliran dengan kondisi 3.

Tabel 8. Hasil pengujian kontrol aliran dengan JST kondisi 3.

No	Set Point (L/h)	Rise time (detik)	Overshoot (L/h)
1	60	2.4	14.196
2	60	2.8	16.912
3	60	1.4	25.794
4	60	2.8	10.784
5	60	2.6	8.343
6	60	3.4	14.088
7	60	4	5.176
8	60	1.8	18.48
9	60	2.6	9.294
10	60	4.2	5.745

Dapat dilihat pada tabel 8, *overshoot* yang besar dan *rise time* yang cepat selalu terjadi pada JST dengan kondisi 3 ini. *Rise time* yang cepat akan cenderung membuat overshoot semakin besar.



Gambar 27. Hasil pengujian pertama kontrol aliran dengan kondisi 4.

Tabel 9. Hasil pengujian kontrol aliran dengan JST kondisi 4.

No	Set Point (L/h)	Rise time (detik)	Overshoot (L/h)
1	60	2.2	15.686
2	60	2	20.922
3	60	1.8	23.657
4	60	2	19.57
5	60	1.2	30.108
6	60	3.6	4.882
7	60	2	21.108
8	60	1.6	22.059
9	60	2.6	13.578
10	60	1	28.098

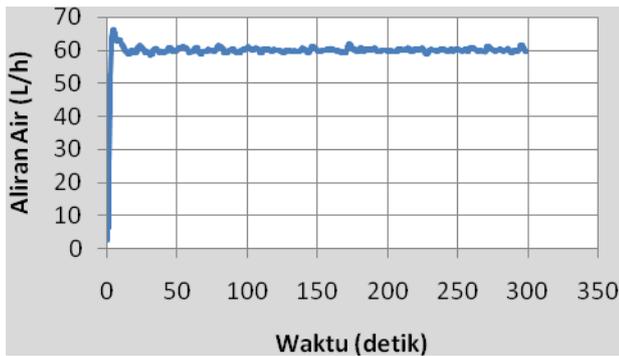
Semakin banyak *hidden layer* yang digunakan, semakin cepat *rise time* yang terjadi dan dengan semakin cepatnya *rise time* ini yang timbul adalah *overshoot* yang semakin besar.

Pengujian yang telah dilakukan ini, membuktikan bahwa JST dalam kontrol aliran air tidak mengalami osilasi secara terus menerus walaupun menggunakan 4 *hidden layer* tidak seperti pada kontrol level air. Jadi tidak ada patokan dalam penggunaan *hidden layer* untuk membuat JST bekerja dengan optimal untuk proses kontrol. Belum tentu penggunaan *hidden layer* dengan jumlah tertentu pada suatu sistem akan berdampak sama dengan sistem lainnya. Yang pasti, dengan semakin banyaknya *hidden layer* yang digunakan menyebabkan JST menjadi semakin kompleks dan membuat JST semakin lama dalam proses belajarnya.

Setelah diuji dengan jumlah *hidden layer* yang berbeda, pengujian selanjutnya adalah pengujian JST dengan menambahkan jumlah *neuron* dari sebelumnya. Jumlah *neuron* yang digunakan adalah 10.

Pengujian selanjutnya dengan kondisi JST:

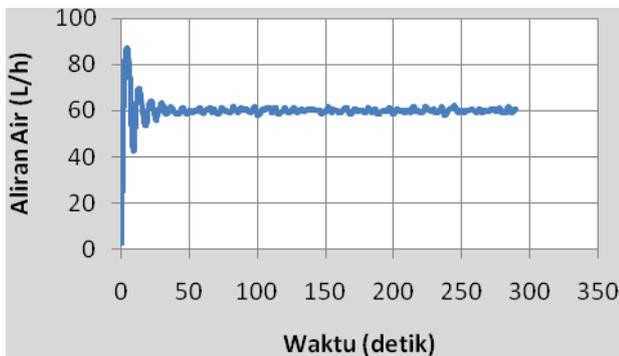
- *Hidden layer* = 1
- *Neuron* = 10
- *Learning rate* = 0.1
- *Input layer* = 1
- *Output layer* = 1
- *Target* = 60 L/h



Gambar 28. Hasil pengujian pertama dengan kondisi 5. Dapat gambar 30 dibandingkan dengan pengujian JST dengan kondisi pertama, JST dengan 10 neuron ini memiliki rise time yang lebih cepat.

Pengujian selanjutnya dengan kondisi JST:

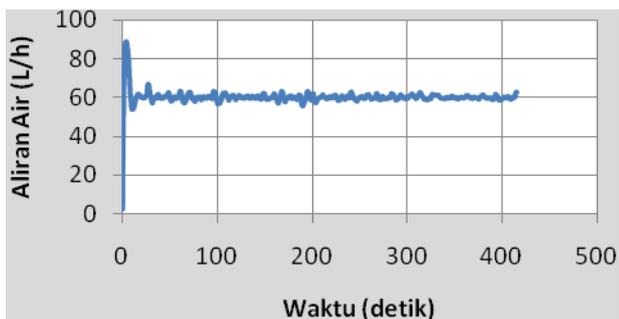
- Hidden layer = 2 - Neuron = 10
- Learning rate = 0.1 - Input layer = 1
- Output layer = 1 - Target = 60 L/h



Gambar 29. Hasil pengujian pertama dengan kondisi 6.

Pengujian selanjutnya dengan kondisi JST:

- Hidden layer = 3 - Neuron = 10
- Learning rate = 0.1 - Input layer = 1
- Output layer = 1 - Target = 60 L/h



Gambar 30. Hasil pengujian pertama dengan kondisi 7.

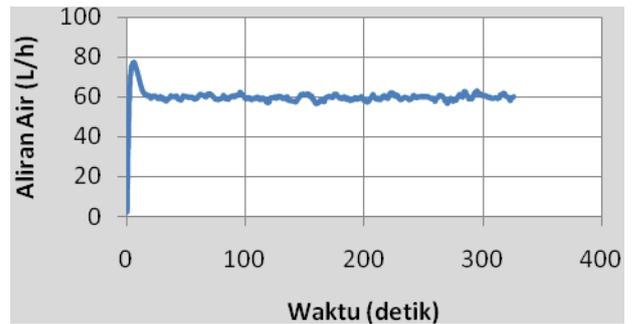
Perbedaan jumlah neuron yang digunakan, akan berpengaruh juga terhadap respon dari kontrol JST tersebut. Dari 3 pengujian yang telah dilakukan dengan jumlah neuron yang ditingkatkan dari 5 menjadi 10, dapat dilihat bahwa semakin banyak neuron yang digunakan, semakin cepat rise

time yang dihasilkan. Seperti pada kondisi 5 pada gambar 26, dibandingkan dengan kondisi 1 yang hanya memiliki 5 neuron. Hasil dengan kondisi 5 tidak lagi landai seperti pada kondisi 1.

• **Pengaruh learning rate terhadap respon sistem**

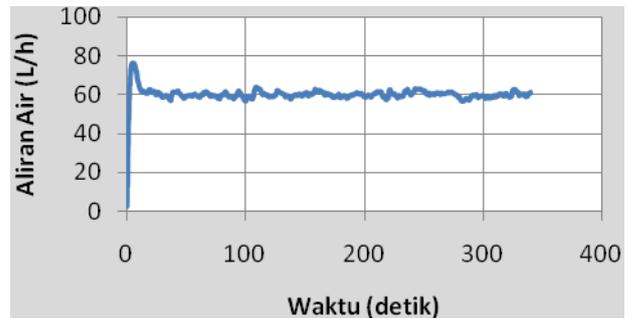
Seperti pada pengujian kontrol level, kontrol aliran air menggunakan JST diuji juga dengan memberikan nilai learning rate yang berbeda. Berikut ini adalah hasil pengujian tersebut.

- Hidden layer = 1 - Neuron = 5
- Learning rate = 0.08 - Input layer = 1
- Output layer = 1 - Target = 60 L/h



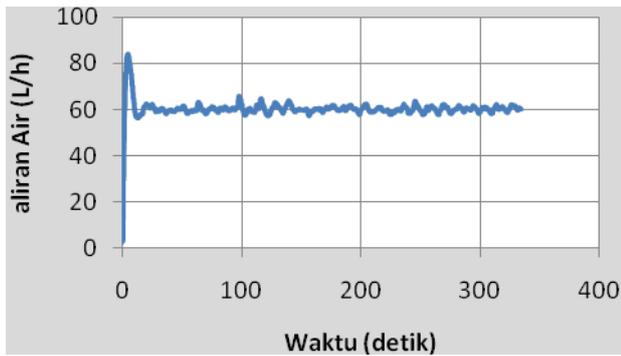
Gambar 31. Hasil pengujian kontrol aliran dengan learning rate 0.08

- Hidden layer = 1 - Neuron = 5
- Learning rate = 0.1 - Output layer = 1
- Input layer = 1 - Target = 60 L/h



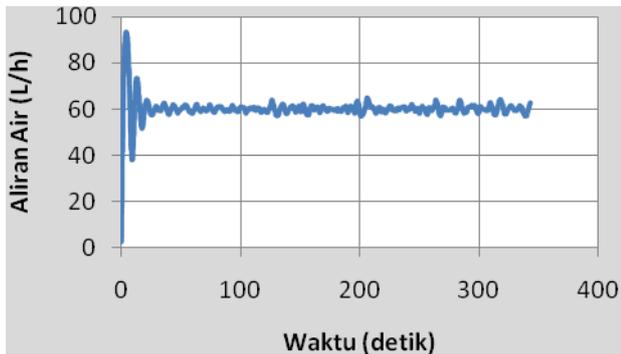
Gambar 32 Hasil pengujian kontrol aliran dengan learning rate 0.1

- Hidden layer = 1 - Neuron = 5
- Learning rate = 0.2 - Input layer = 1
- Output layer = 1 - Target = 60 L/h



Gambar 33. Hasil pengujian kontrol aliran dengan learning rate 0.2

- Hidden layer = 1
- Neuron = 5
- Learning rate = 0.4
- Output layer = 1
- Input layer = 1
- Target = 60 L/h



Gambar 34. Hasil pengujian kontrol aliran dengan learning rate 0.4.

Tabel 10. Hasil pengujian pengaruh learning rate

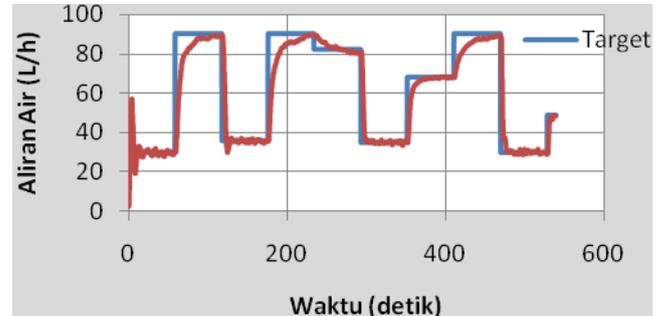
No	Learning rate	Set Point (L/h)	Rise time (detik)	Overshoot (L/h)
1	0.08	60	2.4	77.7451
2	0.1	60	2.4	76.3137
3	0.2	60	2.2	83.7941
4	0.4	60	2	93.2745

Dari tabel 10 dan gambar respon sistem, penambahan nilai dari *learning rate* akan berpengaruh terhadap kecepatan dari respon sistem. Pada kontrol aliran air ini, semakin besar *learning rate* yang digunakan, *rise time* yang dihasilkan semakin cepat. Tetapi dengan cepatnya *rise time* ini, menimbulkan *overshoot* dan adanya osilasi awal. Jadi semakin besar nilai dari *learning rate* semakin besar pula *overshoot* yang dihasilkan. Berbeda halnya dengan kontrol level air sebelumnya. Kontrol level air membutuhkan waktu yang lebih lama daripada kontrol air, karena proses kenaikan air membutuhkan waktu. Sehingga dengan *learning rate* yang besar tidak timbul osilasi dan *overshoot* seperti yang dihasilkan pada kontrol aliran air. Dalam kontrol level air, JST memiliki waktu yang lebih lama, sehingga dengan *learning rate* yang besar membuat JST cepat belajar dan tidak timbul *overshoot* yang besar. Sedangkan dalam kontrol aliran air, tegangan yang diberikan ke pompa berhubungan langsung dengan kecepatan aliran air yang dihasilkan. Hal ini yang membuat JST dengan *learning rate* yang tinggi akan

menghasilkan *rise time* yang cepat, *overshoot* yang besar dan osilasi pada awal pembelajaran.

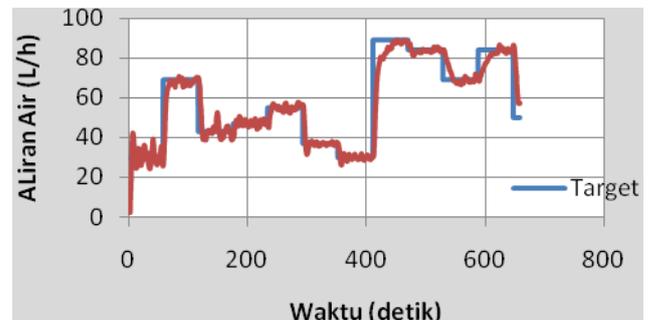
Pengujian selanjutnya adalah pengujian JST dengan memberikan perubahan target aliran air secara acak dan periodik. Perubahan target diberikan setiap 1 menit sekali atau 60 detik. Pengujian ini dilakukan untuk mengetahui apakah JST dapat merespon secara baik dengan perubahan target yang diberikan.

1.1 hidden layer dan 5 neuron



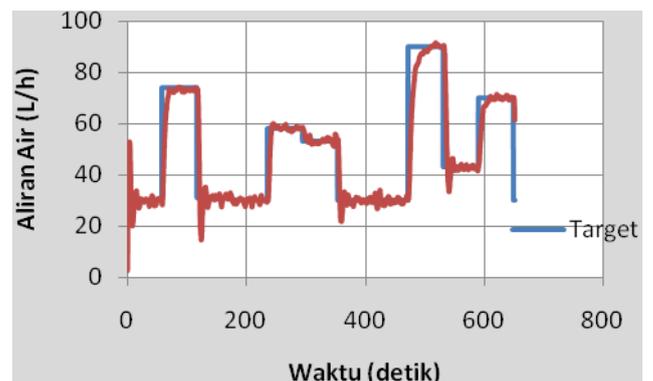
Gambar 37. Hasil pengujian JST dengan target yang berubah-ubah.

2.2 hidden layer dan 5 neuron



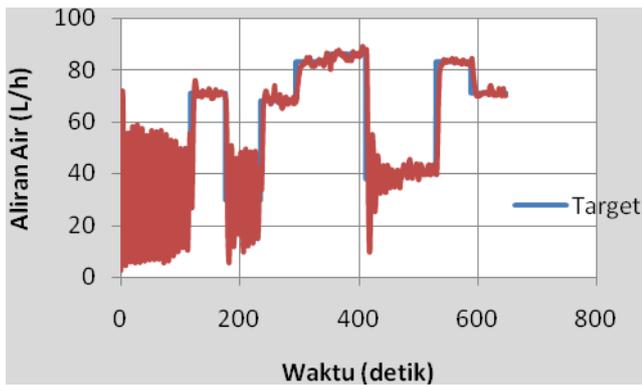
Gambar 38. Hasil pengujian JST dengan target yang berubah-ubah.

3.3 hidden layer dan 5 neuron

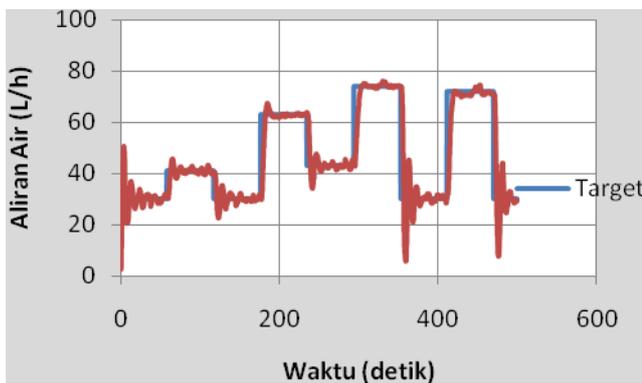


Gambar 31. Hasil pengujian JST dengan target yang berubah-ubah.

4.4 hidden layer dan 5 neuron



Gambar 32. Hasil pengujian JST 4 hidden layer dan 5 neuron dengan target yang berubah-ubah
5.5 hidden layer dan 5 neuron



Gambar 33. Hasil pengujian JST 5 hidden layer dan 5 neuron dengan target yang berubah-ubah.

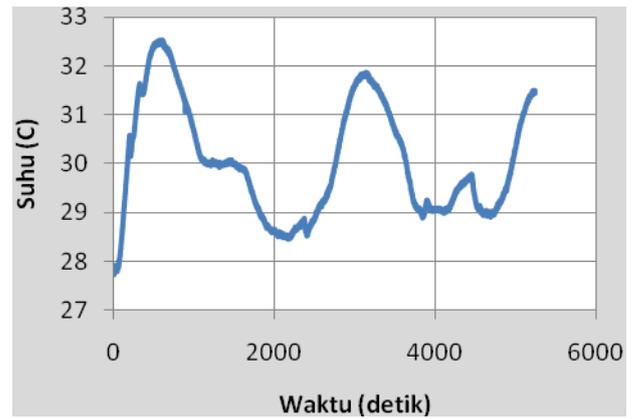
5 hasil data pengujian kontrol aliran dengan perubahan target secara acak dan periodik menggambarkan bahwa JST mampu terus belajar dengan perubahan yang ada. Dari 5 pengujian tersebut, semakin kita ketahui bahwa semakin banyak *hidden layer* yang digunakan, *rise time* yang dihasilkan semakin kecil tetapi menimbulkan *overshoot* yang semakin besar dan semakin lama dalam proses pembelajaran.

3) Kontrol suhu menggunakan JST

Pengujian selanjutnya adalah menguji perangkat lunak Jaringan Saraf Tiruan sebagai kontrol suhu. Waktu cuplik yang digunakan adalah 200 ms. Aktuator pada pengujian ini adalah *heater* yang diberikan tegangan antara 0-10 volt. Heater akan memanaskan air yang berada pada tangki proses, dan perubahan suhu akan dibaca oleh sensor PT 100.

Kondisi JST dalam kontrol suhu adalah sebagai berikut:

- *Hidden layer* = 1
- *Neuron* = 5
- *Learning rate* = 0.1
- *Input layer* = 1
- *Output layer* = 1
- *Target* = 30 celcius



Gambar 34. Hasil pengujian kontrol suhu dengan JST
Untuk menaikkan suhu, dibutuhkan waktu yang cukup lama. Hal ini yang membuat JST sulit untuk mengontrol suhu tersebut dikarenakan JST memerlukan pembelajaran terlebih dahulu secara online sehingga untuk mencapai target, JST membutuhkan waktu yang lebih lama lagi.

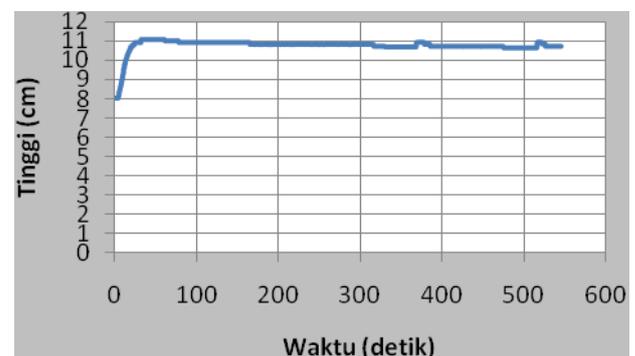
4) Proses kontrol menggunakan PID

Proses kontrol menggunakan PID ini digunakan sebagai pembandingan proses kontrol menggunakan JST. PID yang digunakan adalah PID yang telah ada pada modul DL 2314. Pengaturan K_p , K_i , dan K_d nya dilakukan secara manual pada modul tersebut.

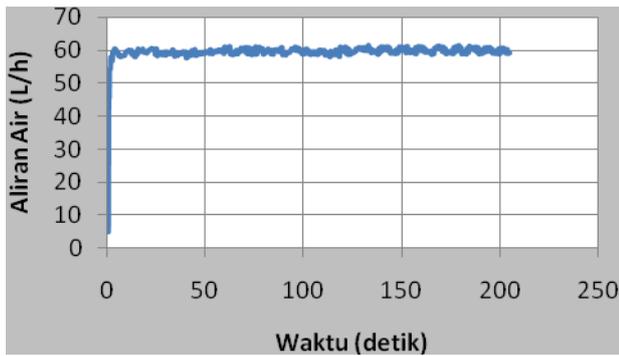
Sistem akan dibuat sebagai berikut:

- Delivery valve → open
- Motor valve → open
- Sol valve → open (On)
- Man valve → open
- Air valve → open
- Drain valve → closed
- Needle valve → closed
- Process Tank → 8 cm

Pengambilan data pertama adalah kotrol level air dengan menggunakan PID sebagai kontrolernya. Kondisi awal adalah 8 cm dengan target yang akan dicapai adalah 11 cm atau sekitar 5.9V. Nilai K_p adalah 25%, K_i 50%, dan 0% K_d . Dengan *range* output pompa adalah 0-10 V.



Gambar 35. Grafik kontrol Level menggunakan PID



Gambar 36. Kontrol aliran air dengan PID

Gambar 36 adalah hasil pengujian kontrol level air dengan PID sebagai kontroler. Dari pengujian ini didapatkan:

Rise time = 30 detik.

Max overshoot = 0.1 cm

Sedangkan pada gambar 4.51 adalah pengujian kontrol aliran air dengan PID yang telah ada pada modul De Lorenzo. Data aliran air di cuplik setiap 200 ms. Target yang diinginkan adalah 60 L/h. *Tuning Kp, Ki, dan Kd* dilakukan secara manual seperti pada kontrol level sampai tercapai kombinasi terbaik. Dari kontrol aliran air menggunakan PID sebagai kontroler, didapatkan:

Rise time = 2.4 detik.

Max overshoot = 1 L/h.

Dari pengujian ini, dapat disimpulkan bahwa kontrol PID lebih baik dari pada kontrol menggunakan JST backpropagation online. Hal ini disebabkan proses kontrol JST online membutuhkan waktu untuk proses belajar ketika proses kontrol dilakukan.

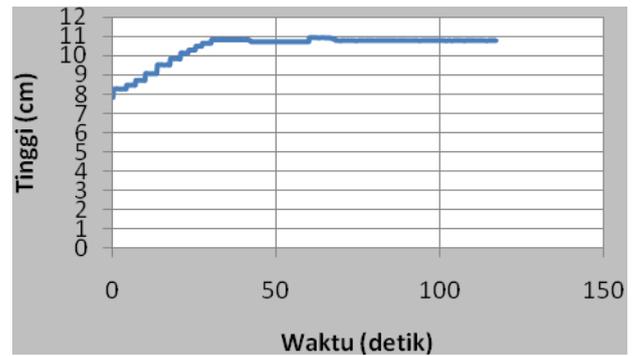
5) Proses kontrol menggunakan Genetik Algoritma (GA)

Pengujian ini dilakukan dengan menggabungkan antara kendali PID dengan metode genetika algoritma. Genetik algoritma akan menghasilkan nilai K_p , K_i , K_d yang baik untuk sistem kontrol PID tersebut. Nilai konstanta ini dihasilkan dalam proses *offline*.

Pengujian pertama adalah kontrol level air dengan target 11 cm. Pengujian kendali level dilakukan dengan memberikan nilai parameter algoritma genetika sebagai berikut :

1. Populasi = 10
2. Konstanta Crossover = 75
3. Konstanta Mutasi = 2
4. Rentang pencarian parameter K_p = 0 sampai 500
5. Rentang pencarian parameter K_i = 0 sampai 100
6. Rentang pencarian parameter K_d = 0 sampai 100

Dengan menggunakan GA, didapatkan $K_p=210.998$, $K_i=0.002$, dan $K_d= 64.704$. Capaian generasi yang didapatkan keluar dengan kondisi generasi yang ke-78.



Gambar 37. Grafik kontrol level menggunakan optimasi PID dengan GA

Dari hasil pengujian ini, didapatkan respon sistem kontrol level menggunakan GA dengan target 11 cm adalah sebagai berikut:

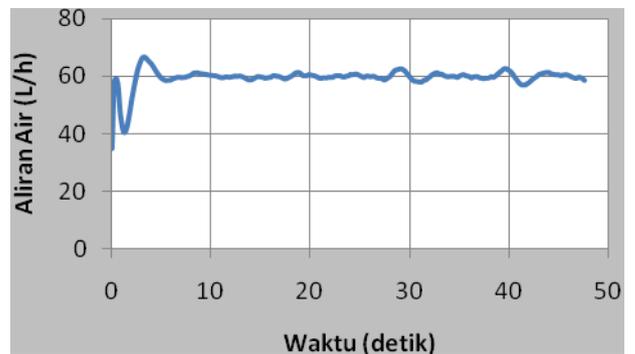
Rise time = 30 detik

Max overshoot = --

Errro steady state = 0.25 cm

Nilai K_p , K_i , dan K_d yang dihasilkan oleh GA bukan terbaik, tetapi dapat digunakan sebagai nilai K_p , K_i , dan K_d yang baik bagi sistem.

Pengujian selanjutnya adalah pengujian kontrol aliran air dengan target 60 L/h. Pengujian dilakukan dengan memberikan waktu sampling 200 ms pada kendali PID. GA menghasilkan $K_p= 0.4535$, $K_i= 0.097$, dan $K_d= 0.369$. Capaian generasi yang didapatkan keluar dengan kondisi generasi yang ke-25.



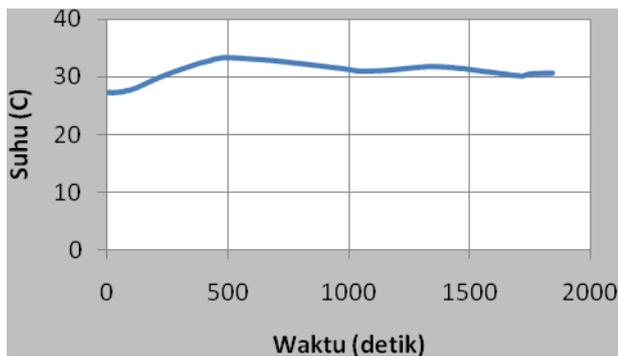
Gambar 38. Grafik kontrol aliran air menggunakan optimasi PID dengan GA

Dari hasil pengujian ini, didapatkan respon sistem kontrol aliran air menggunakan GA dengan target 60 L/h adalah sebagai berikut:

Rise time = 2.6 detik

Max overshoot = 4.48 L/h

Pengujian selanjutnya adalah pengujian kontrol suhu dengan target 30 °C. Pengujian pada suhu dilakukan dengan mengisi tangki pada ketinggian 10 cm. Ketika pengendalian dimulai, maka solenoid valve akan terbuka dan pompa akan diberikan tegangan konstan 2.9 volt. Populasi algoritma genetika 10, dengan probabilitas pindah silang 75 persen, dan mutasi 2 persen. Parameter yang digunakan, $K_p=2.655$, $K_i=0.907$, $K_d=3.898$.



Gambar 39. Grafik kontrol suhu air menggunakan optimasi PID dengan GA
 Dari hasil pengujian ini, didapatkan respon sistem kontrol suhu air menggunakan GA dengan target 30 °C adalah sebagai berikut:

Rise time = 260 detik
Max overshoot = 3.438 °C

Dari hasil ketiga metode kontrol yang telah diuji, sistem yang terbaik adalah sistem PID dengan tuning manual. Nilai K_p , K_i dan K_d didapatkan dengan *trial and error*. Sehingga nilai konstanta yang menghasilkan respon terbaiklah yang akan dipilih secara manual. Tetapi dengan metode optimasi PID dengan GA, nilai konstanta PID yang baik dapat dicari. Walaupun nilai konstanta yang didapat bukan yang terbaik, tetapi nilai ini bisa dianggap dapat diterima (*acceptable*).

Sedangkan untuk sistem kontrol JST backpropagation yang menggunakan metode online, mampu bekerja dengan baik di kontrol aliran air dan level. Walaupun JST membutuhkan waktu yang lebih lama untuk mencapai target yang diinginkan.

V. KESIMPULAN

Pada proyek akhir ini telah melakukan pengendalian level air, aliran dan suhu menggunakan JST backpropagation online sebagai kontroler. Berdasarkan data hasil pengujian dan analisa dapat diambil kesimpulan sebagai berikut:

1. Dalam proses kontrol level air, JST dapat bekerja maksimal ketika JST menggunakan 1 atau 2 *hidden layer*. Ketika JST menggunakan 4 *hidden layer*, JST mengalami osilasi sehingga tidak dapat mencapai *steady state*. Hasil pengujian dapat dilihat pada tabel 4.
2. Besar kecilnya *learning rate* yang digunakan, akan mempengaruhi kinerja JST. Semakin besar nilai dari *learning rate* maka semakin cepat respon sistem untuk mencapai target.
3. Dalam pengujian kontrol suhu, JST membutuhkan waktu yang lebih lama dari pada PID. JST membutuhkan pembelajaran terlebih dahulu sedangkan kenaikan suhu dan penurunan suhu membutuhkan waktu yang lama juga.
4. Jika dibandingkan dengan kontrol konvensional seperti PID, PID dapat bekerja maksimal dalam proses kontrol jika mendapatkan nilai kombinasi K_p , K_i dan K_d yang

sesuai, dan dengan sistem yang berbeda, nilai K_p , K_i dan K_d juga akan berbeda, jadi harus diatur secara manual. Sedangkan JST mampu belajar secara online tanpa perlu mengetahui sistem tersebut.

5. GA dapat digunakan sebagai pengganti *tuning manual* dalam pencarian konstanta PID. Dengan GA dapat menghasilkan nilai K_p , K_i , K_d yang baik, tetapi bukan yang terbaik. Dengan kata lain, nilai konstanta yang dihasilkan, *acceptable* atau dapat diterima.

DAFTAR PUSTAKA

- [1] Kuswadi, Son. 2007, "Kendali Cerdas, Teori dan Aplikasi Praktisnya", Surabaya; Andi
- [2] Puspita, Eru. 2004, "Slide Ajar Kuliah Kontrol Cerdas D4 Elektronika – Kontrol Cerdas", Surabaya
- [3] Puspita, Eru. 2004, "Slide Ajar Kuliah Kontrol Cerdas D4 Elektronika – NN", Surabaya
- [4] Puspita, Eru. 2004, "Slide Ajar Kuliah Kontrol Cerdas D4 Elektronika – Back Propagation", Surabaya
- [5] Puspita, Eru. 2009, "Kontrol Cerdas D4 Elektronika Update Bobot NN", Surabaya.
- [6] Puspita, Eru. 2009, "Kontrol Cerdas D4 Elektronika Contoh Program NN", Surabaya.
- [7] Adam Series. 2007, "ADAM-5000 Series RS-485 Based Data Acquisition and Control System User's Manual"
- [8] Modul Praktikum Kontrol Cerdas, Praktikum V – Jaringan Saraf Tiruan
- [9] http://www.delorenzogloball.com/index.php?option=com_content&view=article&id=689%3Adl-2314&catid=160&Itemid=100102&lang=en
- [10] <http://creasoft.wordpress.com/2008/04/21/jaringan-saraf-tiruan-artificial-neural-network/>