

PERANCANGAN KONTROLER SELF TUNING NEURO-PI UNTUK PENGENDALIAN KETINGGIAN AIR DAN TEMPERATUR STEAM DRUM BOILER

Dennis Tejawijaya^{*)}, Aris Triwiyatno, and Budi Setiyono

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: dennistejawijaya@gmail.com

Abstrak

Boiler dalam sistem Pembangkit Listrik Tenaga Uap (PLTU) memiliki beberapa sistem, salah satunya Steam Drum yang terdiri dari dua buah subsistem; yaitu ketinggian air dan temperature uap. Pengendalian steam drum ini diperlukan untuk menjaga agar steam drum menghasilkan produk atau keluaran yang diinginkan tanpa menyebabkan potensi-potensi kerusakan. Pada Tugas Akhir ini dilakukan penelitian tentang perancangan kontroler Self Tuning Neuro PI yang mengimplementasikan metode Neural Network yang beradaptasi dengan sistem pembelajaran backpropagation berdasarkan data masukan dan keluaran yang telah diinisialisasikan, sehingga didapat bobot-bobot jaringan yang sesuai dengan model data masukan dan keluaran pada pembelajaran. Kemampuan Neural Network ini digunakan untuk tuning parameter P dan I pada kontroler PI agar mampu bekerja secara optimal walaupun terjadi gangguan. Hasil simulasi disimpulkan bahwa dengan menambahkan Neural Network sebagai metode Self Tuning pada kontroler PI dapat mengurangi overshoot sebesar 4.3% dengan nilai IAE sebesar 2,84 pada ketinggian air, sedangkan pada sistem temperatur uap mempercepat rise time dan settling time 7,5 menit, menghilangkan overshoot dan mengurangi IAE sebesar 28,96%. Selain itu, Self Tuning Neuro PI dapat bekerja dengan baik dalam mempertahankan nilai berada pada nilai yang diinginkan meskipun terjadi perubahan set point dan adanya gangguan.

Kata kunci: Steam drum boiler, Self Tuning Neuro-PI, Sstem ketinggian air, Sistem temperatur uap

Abstract

Boiler in the system of Steam Power Plant consists of some systems, one of them are Steam Drum which has two subsystems; water level and steam temperature. The stability of those systems must be controlled to keep the steam drum to produce the desired results without causing potential damage. In this final project carried out research to build a Self Tuning Neuro PI controller, using Neural Network method which can adapts by backpropagation learning system based on inputs and outputs data that have been initialized. Therefore, the weights of the network appropriate with the inputs and outputs model can be obtained. This ability of Neural Network is used for tuning the P and I parameters in PI controllers so it can work optimally despite a disturbance occured. The simulation results obtained with Self Tuning Neuro PI controller for water level can reduce the overshoot of 4.3% with IAE value is 2.84 while for the steam temperature system accelerate the rise time and settling time to 7.5 minutes, eliminating overshoot and reduces IAE 28,96%. In addition, Self Tuning Neuro PI can work properly because it can keep the output at desired value despite the set point changes and there are disturbances.

Keywords: Steam drum boiler, Self Tuning Neuro-PI, Water Level system, Steam Temperatur system

1. Pendahuluan

Boiler atau ketel uap adalah suatu alat berbentuk bejana tertutup yang digunakan untuk menghasilkan steam. Sistem boiler terdiri dari: sistem air umpan (feed water system), sistem steam (steam system) dan sistem bahan bakar (fuel system). Sistem air umpan (feed water system) menyediakan air untuk boiler secara otomatis sesuai

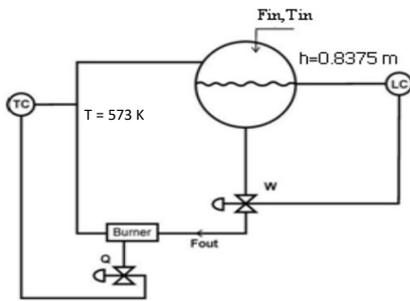
dengan kebutuhan steam.[1]. *Steam Drum* adalah komponen penting untuk jenis *boiler* secara alamiah, *boiler* bertekanan, dan *boiler* dengan gabungan sirkulasi. Fungsi *steam drum boiler* adalah mencampur air yang masuk ke dalam drum (*feedwater*) dengan air yang tersirkulasi dalam boiler, menyediakan air untuk dihantarkan ke komponen penguap (*evaporator*) melalui pipa aliran turun (*downcomer*), Menerima campuran air/uap panas (steam) yang berasal dari pipa aliran naik

(riser), dan memisahkan air dan uap panas. Ketinggian air pada *steam drum* adalah hal yang penting dan sulit untuk diukur dan dikendalikan. Suatu kendali ketinggian air dalam *drum* yang tepat sangat diperlukan. Ketinggian air yang terlalu tinggi dapat menyebabkan air terbawa masuk ke dalam pipa uap panas (*steam tube*). ketinggian air yang terlalu rendah menyebabkan air di dalam drum tidak dapat mendinginkan pipa pembakaran (*furnace*) yang berpotensi menimbulkan kerusakan [2]. Penelitian-penelitian terhadap *steam drum boiler* memberikan respon hasil yang berbeda-beda bergantung kepada kontroler dan metode kontrol yang digunakan [3]–[7]. Variasi hasil analisis dari penelitian sebelumnya membuat *steam drum boiler* menjadi suatu *plant* yang menarik perhatian untuk diteliti dengan metode kontrol lain, seperti *Self Tuning Neuro PI* [8], [9].

2. Metode

2.1. Pemodelan Sistem *Steam Drum Boiler*.

Steam drum merupakan suatu alat pada boiler untuk menampung air dalam volume besar dan memisahkan fluida antara fase gas dan fase cair setelah pemanasan yang terjadi di dalam boiler. Model matematika dari *steam drum boiler* dibentuk oleh dua variabel yaitu ketinggian air dan temperatur uap [10].



Gambar 1. Sistem *Steam Drum Boiler*[10]

Model matematika dari gambar diatas adalah:[11]

$$F_{out} = k w \sqrt{h} \quad (1)$$

$$Ah \frac{dT}{dt} = F_{in}(T_{in} - T) + \frac{Q}{\rho c_p} \quad (2)$$

$$\text{Dengan } F_{out} = k w \sqrt{h} \quad (3)$$

$$Q = GVC q \quad (4)$$

Model *steam drum boiler* yang digunakan dalam penelitian ini merupakan model yang telah digunakan pada penelitian sebelumnya. Parameter sistem ditampilkan pada Tabel 1.

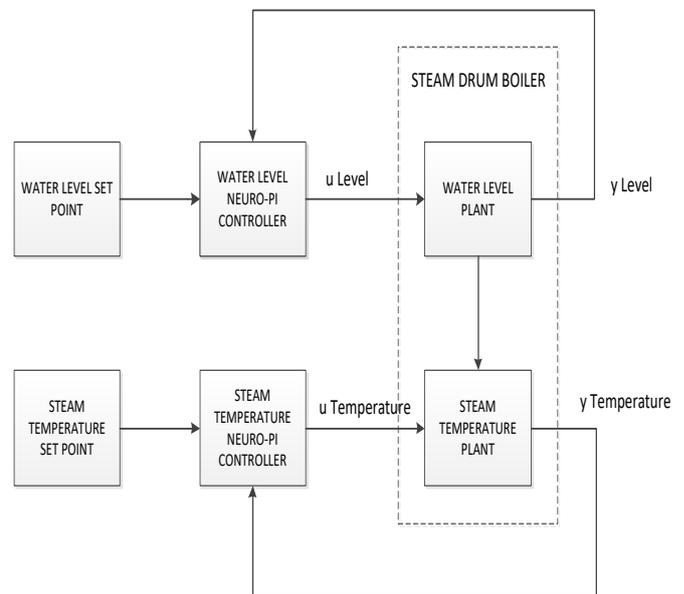
Dengan F_{in} adalah *Flow* air yang masuk (m^3 /menit); F_{out} adalah *Flow* air yang keluar (m^3 /menit); T adalah Temperatur uap (K); T_{in} adalah Temperatur air yang masuk (K); Q adalah *Heat output burner* (Joule/menit); V adalah Volume air (m^3); A adalah Luas *steam drum boiler* (m^2); h adalah Ketinggian air (m); ρ adalah Massa jenis air

(kg/m^3); C_p adalah Kapasitas panas dalam *steam drum* (J/kg K); w adalah *control valve flow* air (m^2); k adalah koefisien *control valve* ($m^{1/2}$ /menit); GVC adalah *Gross Caloric of Coal* (Joule/Kg); dan q adalah Jumlah konsumsi bahan bakar (Kg/menit)

Tabel 1 Parameter fisik model

Parameter	Nilai	Satuan
F_{in}	0,25	m^3 /menit
q	200	Kg/menit
k	1	$m^{1/2}$ /menit
T_{in}	323	Kelvin (K)
Q	Variabel Kontrol	Joule/menit
W	Variabel Kontrol	m^2
T	573	Kelvin (K)
A	2,202415625	m^2
h	0,8375	m
GCV	250.000	J/Kg
ρ	1.000	kg/m^3
C_p	79,676	J/kg K

Perancangan sistem secara keseluruhan dibagi menjadi 2 subsistem, yaitu, subsistem *plant steam drum boiler* dan subsistem kontroler *Neuro-PI*. Parameter tak terikat dalam sistem ini adalah nilai set point ketinggian air dan set point dari temperatur uap, sedangkan variabel terukur adalah sinyal keluaran ketinggian air dan temperatur uap dari *plant steam drum boiler* yang kemudian diumpanbalikan ke masing-masing kontroler.



Gambar 2. skema sistem secara keseluruhan

2.2. Kontroler PI

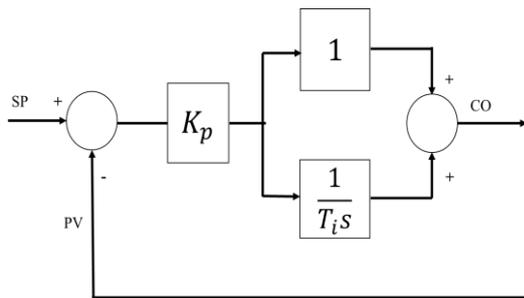
Kontroler proporsional integral biasanya digunakan untuk pengendalian proses yang memiliki dinamika relatif cepat (seperti aliran, tekanan, dan level). Berdasarkan sebuah survei dinyatakan bahwa hampir 80% kontroler PID yang

terpasang di industri menggunakan kontrol PI dalam operasinya. Kontroler PI banyak digunakan pada proses di industri karena permasalahan yang terjadi telah dapat diatasi secara maksimal meskipun hanya dengan kontroler PI. Persamaan matematis dari kontroler PI adalah: [12]

$$CO(t) = K_p \cdot [e(t) + \frac{1}{T_i} \int_0^t e(t) dt] \quad (5)$$

Dalam kawasan Laplace, persamaan 2.18 tersebut dapat direpresentasikan:

$$CO(s) = K_p \cdot [1 + \frac{1}{T_i s}] e(s) \quad (6)$$



Gambar 3. Struktur kontroler PI [12]

Keterangan:

CO = Kontroler Output

Kp = Kontroler Proporsional

e = Error

Ti = Waktu integral

PV = Process Variable

2.3. Kontroler Neuro PI

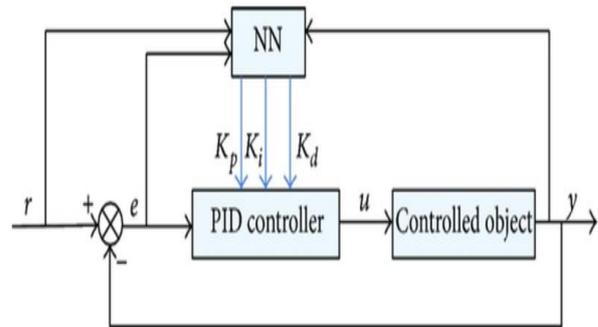
Kontroler PI tidak selalu menghasilkan solusi umum untuk seluruh masalah pengendalian, terutama jika proses tersebut bersifat kompleks, *time-variant*, dengan *delay* dan non-linear [4]. Dalam rangka untuk mencapai kinerja kontrol yang baik, salah satunya adalah menggunakan metode kontrol *Neural Network self-tuning* PI. Metode ini menggabungkan kontroler PI konvensional dan algoritma *Neural Network*. *Neural Network* mengatur *gain* KP dan KI dari kontroler PI dalam prosedur *on-line*. Dengan demikian, keterbatasan *tuning* kontroler PI menggunakan metode *tuning* klasik dapat diatasi.

Desain kontroler *Neural Network* dapat menggunakan nilai referensi yang diinginkan, *error*, ataupun keluaran dari plant sebagai masukan untuk *self-tuning*, dan sinyal keluaran ($y(t)$) dari *Neural Network* akan menjadi sinyal untuk *tuning* PI. Sinyal *error* (e) adalah didefinisikan sebagai perbedaan antara yang diinginkan Posisi (θ_r) dan posisi aktual (θ) dari sudut sendi untuk lengan robot. Sinyal keluaran *Neural Network* ditambahkan ke kontroler PID konvensional untuk menyesuaikan *gain* dari kontroler PID sesuai dengan perubahan dari sinyal *error*. Hal ini memastikan bahwa *Neural Network* akan memberikan perubahan yang diperlukan dalam sinyal kontrol (u). Aksi kontrol dari kontroler PID setelah menggunakan sinyal

keluaran *Neural Network* ($y(t)$) dapat dideskripsikan sebagai berikut:

$$u = K_{PNN} e(t) + K_{INN} \int e dt + K_{DNN} \frac{de(t)}{dt} \quad (7)$$

Dimana K_{PNN} , K_{INN} , dan K_{DNN} adalah *gain* baru dari kontroler *self-tuning* PID dengan rumus $K_{PNN} = y(t) \cdot K_P$; $K_{INN} = y(t) \cdot K_I$; dan $K_{DNN} = y(t) \cdot K_D = 0$



Gambar 4. Contoh konfigurasi kontroler *Self-Tuning Neuro PID* [13]

2.4. Perancangan Kontroler Neuro-PI dengan algoritma *backpropagation*.

- Langkah 0 : Inisialisasi data input dan target
Pada langkah ini diinisialisasikan data input dan target untuk pelatihan *Neural Network* menggunakan data penelitian sebelumnya [14]. Data masukan (input) untuk pelatihan *Neural Network* adalah nilai *error* dan $\Delta error$, sedangkan data target yang digunakan adalah sinyal *tuning* KP dan KI untuk kontroler PI.

- Langkah 1 : Pembentukan dan pelatihan jaringan *Neural Network* menggunakan MATLAB.

Setelah inisialisasi data input dan target untuk pelatihan, kemudian akan dibentuk sebuah kerangka jaringan saraf dengan jumlah layer dan neuron yang ditentukan. Dalam tugas akhir ini dirancang sebuah sistem kendali *feed forward neural network* yang digunakan untuk prediksi data sinyal kendali pada sistem *time-series nonlinear* menggunakan MATLAB m.file.

- Langkah 2 : Melakukan Pelatihan data untuk menentukan bobot jaringan *feed forward Neural Network*.

Pada langkah ini akan dilakukan pelatihan terhadap masing-masing jaringan *feed forward neural network* dengan jumlah *hidden neuron* pada *hidden layer* yang berbeda antara 8 hingga 16. Parameter yang diperhatikan dalam perbandingan antar jaringan adalah *Mean Square Error* (MSE) dari hasil *training*, *validation*, dan *testing*.

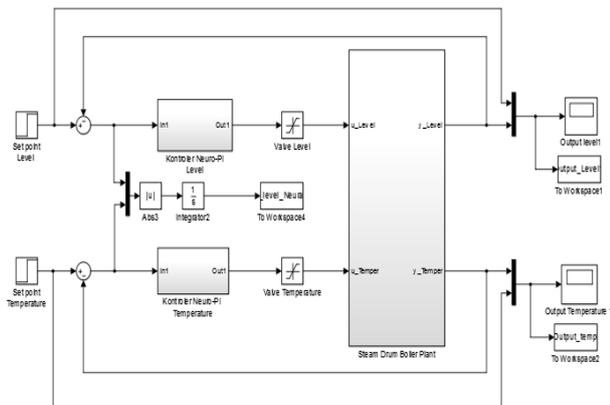
Tabel 2 Perbandingan MSE jaringan *Neural Network* Sistem Ketinggian Air

Hidden neuron	MSE Level ($\times 10^{-4}$)		
	Train	Valid	Test
8	2,66	2,18	2,17
10	1,37	4,69	1,05
12	3,11	1,96	2,01
16	3,42	3,62	2,50

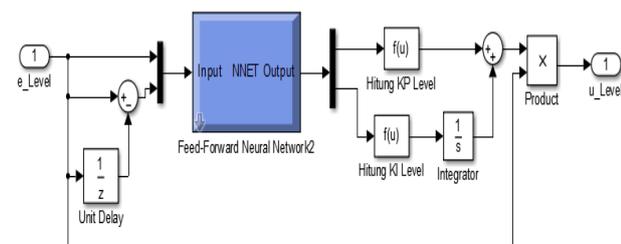
Tabel 3. Perbandingan MSE jaringan *Neural Network* Sistem Temperatur Uap

Hidden neuron	MSE Temperatur ($\times 10^{-4}$)		
	Train	Valid	Test
8	13,96	17,05	16,44
10	14,23	9,81	9,09
12	27,00	16,40	36,39
16	26,33	32,49	29,67

- Langkah 3 : Pembentukan blok Simulink *Neural Network* dan pengaplikasian pada blok simulasi plant. Setelah melakukan pelatihan untuk didapatkan bobot-bobot yang sesuai dengan data pembelajaran (*target*), jaringan akan direpresentasikan dalam blok Simulink agar dapat disimulasikan ke dalam sistem *steam drum boiler* untuk dilakukan pengujian. Pembentukan blok Simulink *Feed Forward Neural Network* pada *command window* MATLAB menggunakan *source code* gensim.



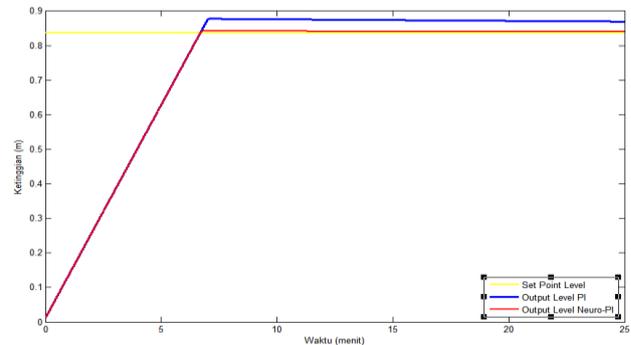
Gambar 5. Rangkaian simulink plant *Steam Drum Boiler* dengan kontroler *Neuro-PI*



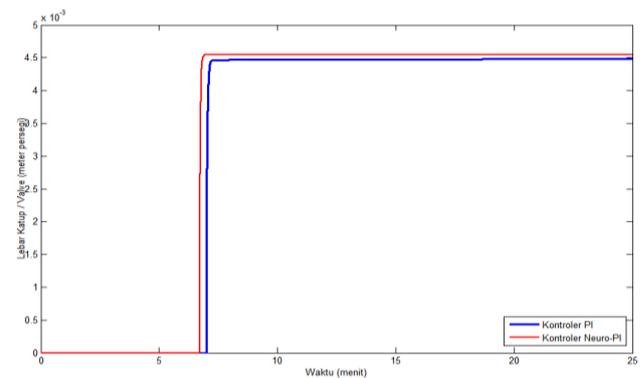
Gambar 6 Rangkaian simulink kontroler *Neuro-PI*

3.1. Hasil dan Analisis Sistem Ketinggian Air

Pengujian terhadap *setpoint* normal dilakukan dengan memberikan masukan sistem berupa sinyal *step* dengan nilai *setpoint* sebesar 0,8375 m. Hasil respon sistem ditampilkan pada gambar 6 dan gambar 7



Gambar 7. Respon ketinggian air *steam drum boiler* SP = 0,8375 m



Gambar 8. Respon kontroler terhadap sistem ketinggian air

Tabel 4. Perbandingan respon sistem pada ketinggian

Respon Sistem	Kontroler Konvensional PI	Kontroler <i>Neuro</i> PI
Rise Time (menit)	5,36	5,36
Settling Time (menit)	7,02	6,365
Maximum Overshoot(%)	4,834	0,537
IAE	3,43	2,844

Tabel 5. Perbandingan respon sinyal kontrol pada ketinggian air

Respon Sinyal Kontrol	PI Ketinggian Air	<i>Neuro</i> PI Ketinggian Air
Action time (menit)	6,98	6,69
Steady state(menit)	8,01	7,25
Energi valve (m ² /menit)	0,004466	0,0045473

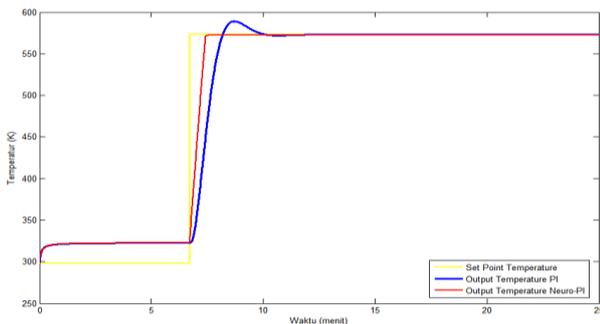
Pengujian dengan set point 0,8375 menghasilkan respon ketinggian air dengan kontroler *Neuro PI* dan konvensional

PI memiliki *rise time* (t_r) yang sama yaitu 5,36 menit. Perbedaan mulai terlihat pada *Settling time* (t_s), dimana kontroler PI membutuhkan waktu 7,02 menit, selisih 0,33 menit lebih lambat dibandingkan PI dengan *Neural Network* yang mencapai *Settling time* saat 6,365 menit. Nilai *overshoot* maksimal pada kontroler PI sebesar 4,834 %, selisih 4,297 % dibandingkan *Neuro PI* yang memiliki *overshoot* maksimal hanya 0,537 %. Nilai IAE pada kontroler *Neuro PI* lebih rendah yaitu 2,844 dibandingkan kontroler PI.

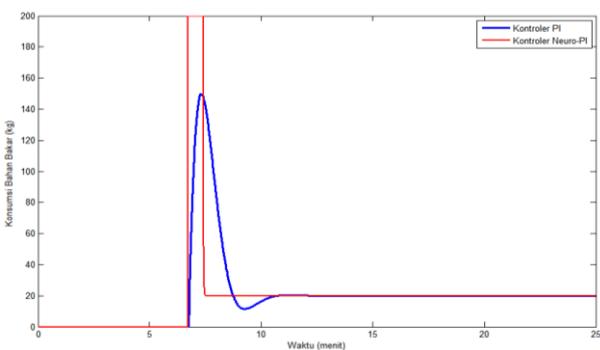
Pengujian respon kontroler *Neuro PI* memberikan respon yang lebih cepat yaitu 6,69 menit, selisih 0,29 menit dengan kontroler PI sebesar 6,98 menit. Respon kontroler *Neuro PI* memiliki waktu *steady* pada menit 7,25, selisih 0,74 menit lebih cepat dibanding konvensional PI yang mencapai *steady* pada menit ke 8,01. Sinyal yang dikirmikan dari kontroler *Neuro PI* memberikan energi yang lebih besar untuk membuka katup sebesar 0,004466 m² per menit sementara konvensional PI sebesar 0,0045473 m² per menit.

3.2. Hasil dan Analisis Sistem Temperatur Uap

Pengujian terhadap setpoint normal dilakukan dengan memberikan masukan sistem berupa sinyal step dengan nilai setpoint sebesar 573 K. Hasil respon sistem ditampilkan pada Gambar 9 dan 10



Gambar 9. Respon temperatur uap steam drum boiler SP = 573 K



Gambar 10. Respon kontroler terhadap sistem temperatur
Tabel 6 Perbandingan respon sistem pada temperatur.

Respon Sistem	Kontroler Konvensional PI	Kontroler Neuro PI
Rise Time (menit)	8,18	7,35
Settling Time (menit)	10,54	7,38
Maximum Overshoot(%)	2,79	0
IAE	361,1	256,5

Tabel 7. Perbandingan respon sinyal kontrol pada temperatur uap

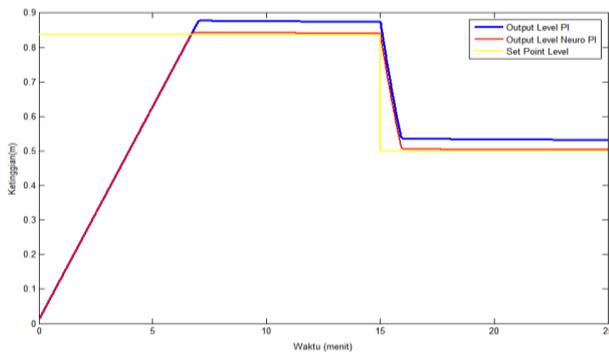
Respon Kontrol	Sinyal	PI Temperatur Uap	Neuro PI Temperatur Uap
Action time (menit)		6,74	6,7
Steady state(menit)		10,75	7,52
Maksimum kebutuhan bahan bakar (Kg)		149,65	200

Hasil respon temperatur dengan kontroler *Neuro PI* mengalami *rise time* (t_r) yang lebih cepat yaitu 7,35 menit, sedangkan kontroler konvensional PI yaitu 8,18 menit. *Settling time* dengan kontroler *Neuro PI* juga lebih cepat yaitu 7,38 menit dibandingkan kontroler PI yaitu 10,54 menit. Nilai *overshoot* maksimal pada kontroler konvensional PI adalah 2,79 % sedangkan kontroler *Neuro PI* tidak memiliki *overshoot*. Nilai IAE pada kontroler *Neuro PI* lebih rendah yaitu 256,5 dibanding kontroler konvensional PI yaitu 361,1.

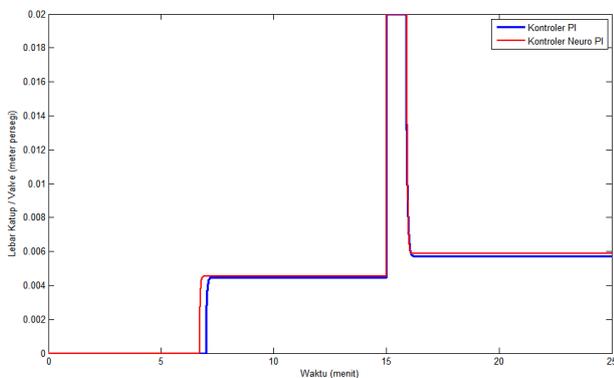
Hasil respon kontroler PI dengan *Neural Network* memberikan *action* dalam waktu yaitu 6,7 menit. Waktu respon yang diberikan kontroler *Self Tuning Neuro PI* ini tidak jauh berbeda dengan kontroler PI sebesar 6,74 menit. Respon kontroler PI dengan logika *Neural Network* memiliki waktu *steady* terhadap *action* sebesar 7,52 menit selisih 3,23 menit lebih cepat dibanding konvensional PI 10,75 menit. Kedua kontroler memberikan sinyal kontrol konsumsi bahan bakar per menit yang hampir sama, tetapi konsumsi bahan bakar maksimum yang dibutuhkan kontroler PI dengan *Neural Network* lebih besar yaitu 200 Kg sementara konvensional PI sebesar 149,65 Kg. kontroler *Self Tuning Neuro PI* memberikan sinyal semaksimal mungkin agar keluaran sistem lebih cepat mencapai *set point*.

3.3. Hasil dan Analisis Terhadap Perubahan Set Point Sistem Ketinggian Air

Pengujian terhadap *setpoint* yang berubah pada sistem ketinggian air dilakukan dengan mengubah *setpoint* sistem ketinggian air menjadi 0,5 m setelah 15 menit sistem berjalan. Perbandingan respon sistem dan respon sinyal kontrol antara *Self Tuning Neuro PI* dan kontroler PI ditunjukkan pada gambar 11.dan 12



Gambar 11. Respon ssstem ketinggian air steam drum boiler dengan perubahan nilai set point pada menit ke - 15



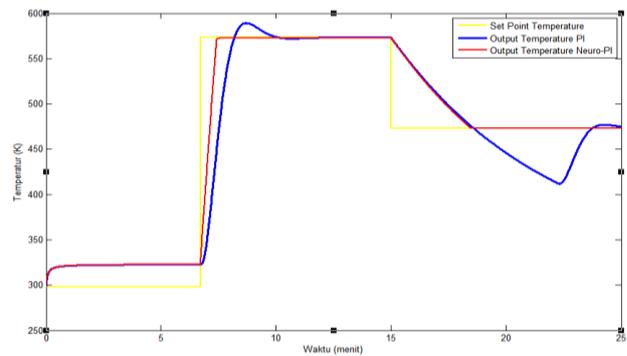
Gambar 12. Respon kontroler terhadap sistem ketinggian air dengan perubahan nilai set point pada menit ke - 15

respon sistem ketinggian air dengan *set point* berubah antara kontroler *Neuto* PI dan kontroler PI memiliki hasil yang sama dengan respon sistem ketinggian air dengan *set point* normal tanpa gangguan. Perbedaan terjadi saat diberikan perubahan *set point* pada menit ke - 15, dimana kontroler *Neuro* PI merespon perubahan *set point* dan mencapai kondisi *steady* terhadap *set point* yang baru setelah 1,08 menit dibandingkan dengan kontroler PI yang *steady* setelah 1,26 menit dan belum mencapai nilai *set point* yang baru.

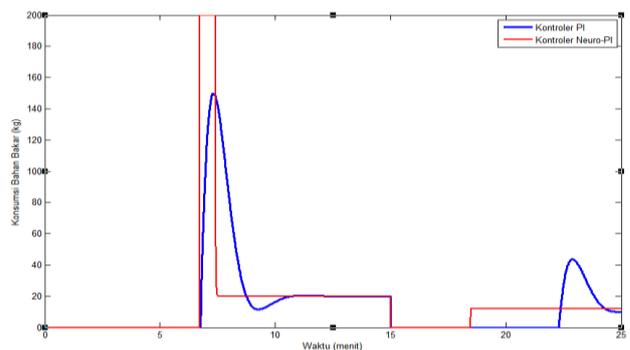
respon sinyal kontrol terhadap sistem ketinggian air dengan *set point* berubah antara kontroler *Neuto* PI dan kontroler PI memiliki hasil yang sama dengan respon sinyal kontrol sistem ketinggian air dengan *set point* normal saat tidak ada gangguan. Saat diberikan perubahan *set point* pada menit ke - 15, kedua kontroler merespon perubahan *set point* dengan besar sinyal yang hampir sama. Pada kondisi *steady*, sinyal kontrol dari *Neuro* PI sebesar 0,059 dan sinyal kontrol dari PI adalah 0,057.

3.4. Hasil dan Analisis Terhadap Perubahan Set Point Sistem Temperatur Uap

Pengujian terhadap *setpoint* yang berubah dilakukan dengan mengubah *set point* sistem level menjadi 473 K setelah 15 menit. Perbandingan respon sistem dan respon sinyal kontrol ditunjukkan pada gambar 13 dan 14



Gambar 13. Respon sistem temperature uap steam drum boiler dengan perubahan nilai set point pada menit ke - 15



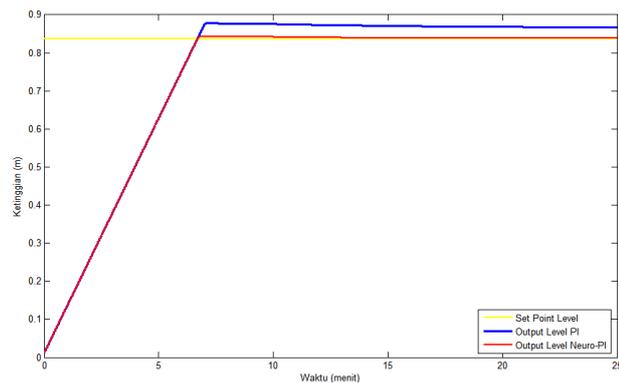
Gambar 14. Respon kontroler terhadap sistem temperature uap dengan perubahan nilai set point pada menit ke - 15

Respon sistem temperatur uap dengan *set point* berubah antara kontroler *Neuto* PI dan kontroler PI memiliki hasil yang sama dengan respon sistem temperatur dengan *set point* normal tanpa gangguan. Perbedaan terjadi saat diberikan perubahan *set point* pada menit ke - 15, dimana kontroler *Neuro* PI merespon perubahan *set point* dan mencapai kondisi *steady* terhadap *set point* yang baru dalam waktu 3,51 menit dibandingkan dengan kontroler PI yang belum mencapai *steady state* dan belum mencapai nilai *set point* yang baru.

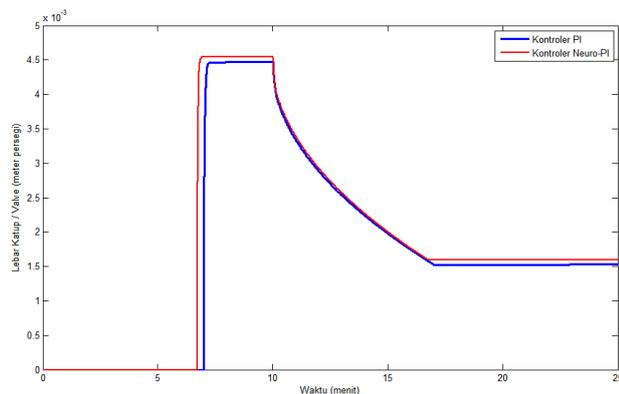
Saat diberikan perubahan *set point* pada menit ke - 15, kedua kontroler merespon perubahan *set point* dengan besar sinyal yang hampir sama. Namun kontroler *Neuro PI* merespon lebih cepat terhadap perubahan sinyal dan memberikan sinyal sebesar 11,9406 pada menit ke 18,52 dan mencapai kondisi *steady*. Sedangkan kontroler *PI* merespon perubahan sinyal lebih lambat dan mulai memberikan sinyal pada menit ke 22,3 dan belum mencapai nilai *steady* hingga proses berakhir.

3.5. Hasil dan Analisis Terhadap Gangguan Sistem Ketinggian Air

Pengujian gangguan pada sistem ketinggian air dilakukan dengan memberikan sinyal gangguan pada sistem seolah-olah terjadi kebocoran pada tangki dengan luas sebesar 0,04 m² yang terjadi pada saat menit ke - 10 saat sistem telah berjalan. Perbandingan respon sistem dan respon sinyal kontrol dari masing-masing kontroler *PI* dan kontroler *Neuro PI* ditunjukkan pada gambar 15 dan gambar 16



Gambar 15. Respon Kontroler Terhadap Gangguan Kebocoran Tangki Sebesar 0,1m² di menit ke-8



Gambar 16. Respon Sinyal Kontrol Terhadap Gangguan Kebocoran Tangki Sebesar 0,1m² di Menit ke - 8

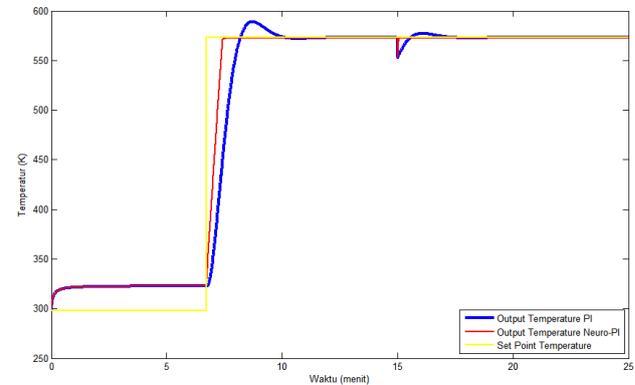
Hasil respon sinyal kedua kontroler memiliki *time rise* yang sama saat pengujian *set point* normal, respon kontroler *Neuro PI* memiliki *action* yang lebih cepat yaitu 6,365 menit, sedangkan konvensional *PI* sebesar 7,02

menit. Perbedaan terjadi saat kebocoran tangki terjadi saat menit ke - 8, kedua kontroler sama - sama memberikan *action* yang sama namun kontroler *Neuro PI* menghasilkan *IAE* lebih minim yaitu 2,844 daripada *IAE* kontroler *PI* yaitu 3,429

Perbedaan respon sinyal kontroler terjadi saat kebocoran tangki terjadi saat menit ke - 10 Kedua kontroler memberikan sinyal yang hampir sama namun kontroler *Neuro PI* memberikan luas katup yang lebih besar yaitu 0,0045473 m² per menit daripada luas katup kontroler *PI* yaitu 0,004466 m² per menit. Kemudian katup menutup secara perlahan saat terjadi gangguan kebocoran, lalu mencapai kondisi *steady* dengan katup kontroler *Neuro PI* sebesar 0,0016 m²/menit pada menit ke 16,51 dan katup dengan kontroler *PI* sebesar 0,0015 m²/menit pada menit ke 16,92.

3.6. Hasil dan Analisis Terhadap Gangguan Sistem Temperatur Uap

Pengujian gangguan pada sistem temperatur uap dilakukan dengan memberikan sinyal gangguan pada sistem seolah-olah terjadi penurunan temperatur pada tangki akibat temperatur lingkungan yang menurun sebesar 20 K yang terjadi pada menit ke - 15 saat sistem telah berjalan. Perbandingan respon sistem dan respon sinyal kontrol ditampilkan pada gambar 17 dan 18

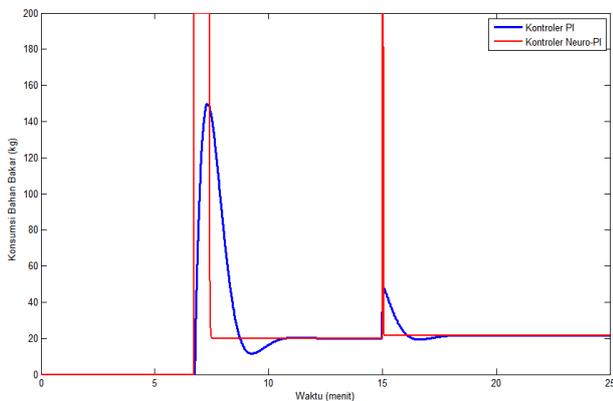


Gambar 17. Respon Keluaran Sistem Temperatur Saat Gangguan Pengurangan Jumlah Bahan Bakar 20Kg/menit Saat menit ke - 15

Respon nilai keluaran kedua kontroler saat terjadi gangguan yang berarti adalah perubahan nilai *IAE* yaitu kontroler *Neuro PI* sebesar 257,6 sedangkan konvensional *PI* sebesar 372,4. Dengan menggunakan kontroler *Neuro PI*, perubahan nilai *IAE* sangat minim saat ada gangguan.

Respon keluaran kontroler *Neuro PI* terlihat mampu dengan cepat memberikan tanggapan saat terjadi gangguan yang terjadi pada menit ke - 20 hingga mencapai nilai *steady* dengan segera. Ini berbeda dengan kontroler

konvensional PI yang membutuhkan waktu sekitar 2 menit untuk menanggapi gangguan dan mencapai nilai *steady*.



Gambar 18. Respon Keluaran Sinyal Kontrol Sistem Temperatur Saat Gangguan Pengurangan Jumlah Bahan Bakar 20Kg/menit Saat menit ke - 15

4. Kesimpulan.

Kontroler *Self Tuning Neuro PI* dapat bekerja dengan baik sebagai pengendalian sistem ketinggian air dan temperature uap pada plant *steam drum boiler*. Hal ini dibuktikan dari hasil uji coba kontroler terhadap plant *steam drum boiler* mampu mengikuti dan menjaga nilai keluaran sistem berada pada *set point* yang diinginkan. Hasil dan analisis terhadap sistem tanpa gangguan, perubahan *set point*, serta dengan adanya gangguan menunjukkan respon keluaran sistem dan sinyal kontrol yang diberikan oleh kontroler *Self Tuning Neuro PI* lebih baik dan lebih cepat dalam memperbaiki nilai *error* dan mencapai *set point* dalam kondisi *steady state*, jika dibandingkan dengan kontroler PI konvensional.

Referensi

- [1] S. A. Muin, *No TiPesawat-Pesawat Konversi Energi I (Ketel Uap)*tle. Jakarta: Rajawali Pers, 1988.
- [2] S. Teir and A. Kulla, "Feedwater and Steam System Components," 2002.
- [3] R. X. Zhao, X. J. Wang, and F. Teng, "The PID control system of steam boiler drum water level based on genetic algorithms," *Proc. 2014 IEEE Chinese Guid. Navig. Control Conf.*, no. 1, pp. 1983–1986, 2014.
- [4] S. Z. S. Al-khayyt, "Tuning PID Controller by Neural Network for Robot Manipulator Trajectory Tracking," vol. 8, no. 1, pp. 19–28, 2013.
- [5] S. R. Valsalam, S. Anish, and B. R. Singh, *Boiler modelling and optimal control of steam temperature in power plants*, vol. 42, no. 9. IFAC, 2009.
- [6] Herlambang, *Desain Pengendalian Ketinggian Air dan Temperatur Uap pada Sistem Steam Drum Boiler dengan Menggunakan Metode Sliding Mode Control (SMC)*. Surabaya: Institut Teknologi Sepuluh November, 2010.
- [7] R. F. Mardhika, A. Triwiyatno, and B. Setiyono, "Perbandingan Kinerja Kontrol Optimal Linear Quadratic Integral Tracking dengan Algoritma Genetika dan Linear Quadratic Tracking untuk Pengendalian Ketinggian Air dan Temperatur Uap Pada Model Steam," *Semarang Fak. Tek. Univ. Diponegoro*, vol. 4, 2015.
- [8] G. Scott, J. Shavlik, and W. Ray, "Refining {PID} controllers using neural networks," *Nips-4*, vol. 4, pp. 555–562, 1992.
- [9] L. Luoren and L. Jinling, "Research of PID Control Algorithm Based on Neural Network," *Energy Procedia*, vol. 13, pp. 6988–6993, 2011.
- [10] M. Metode, L. Quadratic, R. Lqr, E. Apriliani, and M. Si, "Pengendalian optimal pada sistem," pp. 1–10, 2010.
- [11] Stephanopoulos, "Chemical Process Control An Introduction to Theory and Practice," 1984.
- [12] I. Setiawan, *Kontrol PID Untuk Proses Industri*. 2008.
- [13] S. Gao, J. Yang, and J. Wang, "D-FNN Based Modeling and BP Neural Network Decoupling Control of PVC Stripping Process," vol. 2014, 2014.
- [14] A. Pardomuan, *Perancangan Kontroler Self Tuning Fuzzy Pi untuk Pengaturan Ketinggian Air dan Temperatur Uap pada Steam Drum Boiler*. Semarang: Universitas Diponegoro, 2017.