



TUGAS AKHIR - TE 141599

**DESAIN AUTOTUNING KONTROLER PID BERBASIS
ALGORITMA *NEURAL-NETWORK* UNTUK SISTEM
PENGATURAN *CASCADELEVEL* DAN *FLOW LIQUID*
PADA *PLANT COUPLED TANK***

Rheza Qashmal Darmawan
NRP. 2212 100 160

Dosen Pembimbing
Ir. Joko Susila MT.,
Imam Arifin ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE 141599

**DESIGN AUTOTUNING PID CONTROLLER BASED ON
NEURAL-NETWORK ALGORITHM FOR CASCADE
LEVEL AND FLOW LIQUID CONTROL SYSTEM ON
COUPLED TANKS**

Rheza Qashmal Darmawan
NRP. 2212 100 160

Supervisor
Ir. Joko Susila MT.
Imam Arifin ST., MT.

*DEPARTEMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Sepuluh Nopember Insitute of Technology
Surabaya 2016*

**DESAIN AUTOTUNING KONTROLER PID BERBASIS
ALGORITMA NEURAL NETWORK UNTUK SISTEM
PENGATURAN CASCADE LEVEL DAN FLOW LIQUID PADA
PLANT COUPLED TANKS**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I

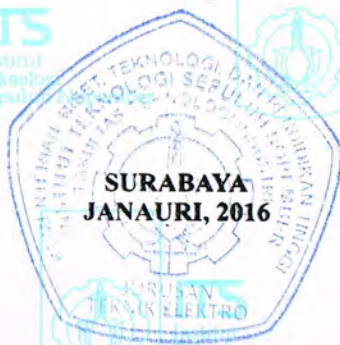
Dosen Pembimbing II

Ir. Joko Susila, MT.

NIP. 19660606199102001

Imam Arifin, ST., MT.

NIP. 197302222002121001



Desain Autotuning Kontroler PID Berbasis Algoritma Neural-Network Untuk Sistem Pengaturan *Cascade Level* dan *Flow Liquid* Pada *Plant Coupled Tanks*

Rheza Qashmal Darmawan
2212 100 160

Dosen Pembimbing I : Ir. Joko Susila, MT.
Dosen Pembimbing II : Imam Arifin, ST., MT.

ABSTRAK

Pada industri proses yang melibatkan fluida, suatu fluida akan dipompa dan dialirkan dari satu tangki ke tangki yang lain untuk diolah. Pemindahan cairan dari satu tangki ke tangki yang lain menyebabkan berubahnya *level* fluida dalam tangki. Dalam pengaturan *level*, pemindahan cairan biasa disebut sebagai pembebanan pada *level*. Perubahan beban ini dapat mempengaruhi dari kinerja kontroler.. Kontroler yang banyak digunakan di industri proses adalah kontroler PID karena kesederhanaan struktur dan kehandalannya. Pada penerapan kontroler PID, *tuning* parameter kontroler sering dilakukan dengan prosedur *trial and error*. Untuk tetap memenuhi spesifikasi kontrol yang diharapkan, maka perlu dilakukan *tuning* ulang parameter kontroler PID. Kontroler PID *Neural Network* ini didesain untuk dapat melakukan *autotuning* pada parameter kontroler PID sehingga dapat mengatasi perubahan parameter pada *plant* dan menjaga performa dari *plant*. Berdasarkan hasil simulasi, sistem pengaturan *level* air pada *plant coupled tank* dengan PID *Neural Network* lebih baik dengan nilai RMSE 0,044 % daripada kontroler PID dengan nilai RMSE 0.35 %. Pada pengujian beban kontroler PI dengan konfigurasi kontrol *cascade* mampu memberikan hasil yang lebih baik dengan nilai RMSE 1.13 %.

Kata Kunci : *Coupled tanks*, kontroler PID, *Neural Network*

Design Autotuning PID Controller based on Neural-Network Algorithm for Cascade Level and Flow Liquid Control System on Coupled Tanks

Rheza Qashmal Darmawan
2212 100 160

Supervisor I : Ir. Joko Susila., MT.
Supervisor II : Imam Arifin ST., MT.

ABSTRACT

In the industrial processes involving fluid, a fluid to be pumped and drained from one tank to another tank for processing. The transfer of liquid from one tank to another led to changes in the liquid level in the tank. In the level control, commonly referred to as the displacement fluid loading level. This load changes may affect the performance of the controller. The main control strategy used is based on the PID controller design because is simple and robustness. In many times the controller tuning is done by trial and error. Plant in the industry can change the parameters that result from changes in the load on the plant. To keep control of who is expected to meet the specifications it is necessary to re-tuning PID controller parameters. Neural Network PID controller is designed to perform autotuning in PID controller parameters so that it can cope with changes in the parameters of the plant and maintain the performance of the plant. Based on simulation results, level control system on plant coupled tank with PID Neural Network is better with RMSE value 0.044% then controller PID with RMSE value 0.35%. For load disturbance testing controller PID with cascade configuration control able to profit to better result with RMSE value 1.13 %.

Keywords :*Coupled tanks, PID Controller, Neural Network*

KATA PENGANTAR

Alhamdulillah Robbil 'Alamin, terucap syukur kehadiran Allah atas limpahan rahmat dan karuninaya serta sholawat serta salam terhadap Nabi besar Muhammad SAW sehingga penulis dapat menyelesaikan Tugas Akhir ini yang berjudul **“Desain Autotuning Kontroler PID Berbasis Algoritma Neural Network untuk Sistem Pengaturan Cascade Level dan Flow Liquid pada Plant coupled tank”** guna memenuhi persyaratan untuk mencapai Gelar Sarjana Teknik pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Dalam kesempatan ini penulis ingin mengucapkan terimakasih kepada pihak-pihak yang banyak berjasa terutama dalam penyusunan Tugas Akhir ini, yaitu:

1. Segenap keluarga besar terutama ibu Dr. Prihat Assih, ayah Agus Dwi Swandaru, dan adik Shabrina Dirmayanti tercinta yang selalu memberi dukungan, semangat, dan doa untuk keberhasilan penulis.
2. Bapak Ir. Joko Susila, MT. dan Bapak Imam Arifin, ST., MT. selaku dosen pembimbing yang telah banyak memberikan saran serta bimbingannya.
3. Bapak Moh. Abdul Hady ST., MT., selaku asisten dosen yang selalu memberi saran dan semangatnya.
4. Seluruh dosen bidang studi Teknik Sistem Pengaturan dan Jurusan Teknik Elektro atas pendidikannya dan ilmunya.
5. Seluruh rekan asisten Lab AJ-104 dan atas dukungan, semangat, bantuan, kebersamaan, dan kerjasamanya selama ini.
6. Seluruh rekan E-52 atas dukungan, kebersamaan dan kerjasamanya selama ini.
7. Serta pihak-pihak yang tidak dapat disebutkan satu persatu atas dukungannya terhadap penulis.

Laporan Tugas Akhir ini masih jauh dari kata sempurna, oleh karena itu penulis mengharapkan kritik dan saran dari pembaca. Semoga buku Tugas Akhir ini dapat memberikan manfaat bagi pembaca sebagai acuan penelitian selanjutnya.

Surabaya, Januari 2016

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
LEMBAR PENGESAHAN	v
ABSTRAK.....	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL.....	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian.....	3
1.5 Metodologi.....	3
1.6 Sistematika Penulisan	4
1.7 Relevansi.....	5
BAB 2 SISTEM PENGATURAN <i>PLANT COUPLED TANK</i>	7
2.1 Sistem Pengaturan.....	7
2.2 Konfigurasi Sistem Pengaturan.....	8
2.2.1 <i>Single Loop</i>	8
2.2.2 <i>Cascade</i>	9
2.3 <i>Coupled Tank</i>	11
2.4 Pemodelan Sistem Pengaturan <i>Level Plant Coupled Tank</i>	12
2.4.1 Pemodelan Pompa.....	13
2.4.2 Pemodelan Pipa.....	13
2.4.3 Pemodelan Tangki	14
2.4.4 Pemodelan Sensor <i>Level</i>	18
2.5 Pemodelan Sistem Pengaturan <i>Flow Plant Coupled Tank</i>	18
2.5.1 Pemodelan Sensor <i>Flow</i>	20
2.6 Kontroler PID [5].....	20
2.7 <i>Disturbances</i> [6].....	21
2.7.1 <i>Load Disturbances</i>	22
2.7.2 <i>Measurement Noise</i>	22
2.8 Jaringan Syaraf Tiruan (<i>Artificial Neural Network</i>) [7]	22
2.8.1 Konsep Dasar Pemodelan <i>Neural Network</i>	23
2.8.2 <i>Learning</i> pada <i>Neural Network</i>	24

2.9	PID – <i>Neural Network</i>	24
BAB 3	PERANCANGAN SISTEM	27
3.1	Gambaran Umum Sistem	27
3.2	Pemodelan Sistem Pengaturan <i>Level Plant Coupled Tank</i>	28
3.2.1	Pemodelan Pompa.....	28
3.2.2	Pemodelan Pipa.....	28
3.2.3	Pemodelan Tangki	28
3.2.4	Pemodelan Sensor <i>Level</i>	29
3.2.5	Fungsi Alih Keseluruhan	29
3.2.6	Pemodelan Sistem Pengaturan <i>Flow Plant Coupled Tank</i>	29
3.3	Perancangan Kontroler.....	30
3.3.1	Perancangan Kontroler PID dengan <i>Root Locus</i>	30
3.3.2	Perancangan Kontroler PID dengan Metode Miluse Viteckova	33
3.3.3	Perancangan Kontroler PID- <i>Neural Network</i>	34
3.4	Perancangan <i>Human Machine Interface</i> (HMI).....	39
BAB 4	PENGUJIAN DAN ANALISA	43
4.1	Simulasi Sistem Tanpa Kontroler	43
4.1.1	Respon Closed Loop Proses <i>Level</i>	43
4.1.2	Respon <i>Open Loop</i> Proses <i>Flow</i>	44
4.2	Simulasi Sistem dengan Kontroler PID	45
4.2.1	Simulasi Pengaturan <i>Level Single loop</i> Variasi <i>Set point</i>	45
4.2.2	Simulasi Pengaturan <i>Level Single Loop</i> Variasi Parameter Kontroler (K_p, τ_i , Dan τ_d)	46
4.2.3	Simulasi Sistem Pengaturan <i>Level Single Loop</i> Dengan Diberi Beban	49
4.2.4	Simulasi Sistem Pengaturan <i>Level Single Loop</i> Dengan <i>Noise</i>	50
4.2.5	Simulasi Sistem Pengaturan <i>Flow Single Loop</i>	51
4.2.6	Simulasi Sistem Pengaturan <i>Level</i> Konfigurasi <i>Cascade</i>	52
4.2.7	Simulasi Sistem Pengaturan <i>Level</i> Konfigurasi <i>Cascade</i> Dengan Diberi <i>Disturbance</i>	53
4.3	Simulasi Sistem dengan Kontroler PID <i>Neural-Network</i>	54
4.3.1	Simulasi Pengaturan <i>Level</i> Air Pada Tangki 2.....	55
4.3.2	Simulasi Dengan Variasi <i>Learning Rate</i>	56
4.3.3	Simulasi Sistem Pengaturan <i>Level</i> Dengan Pemberian Beban.....	57
BAB 5	PENUTUP	61
5.1	Kesimpulan	61
5.2	Saran	61
	DAFTAR PUSTAKA	63

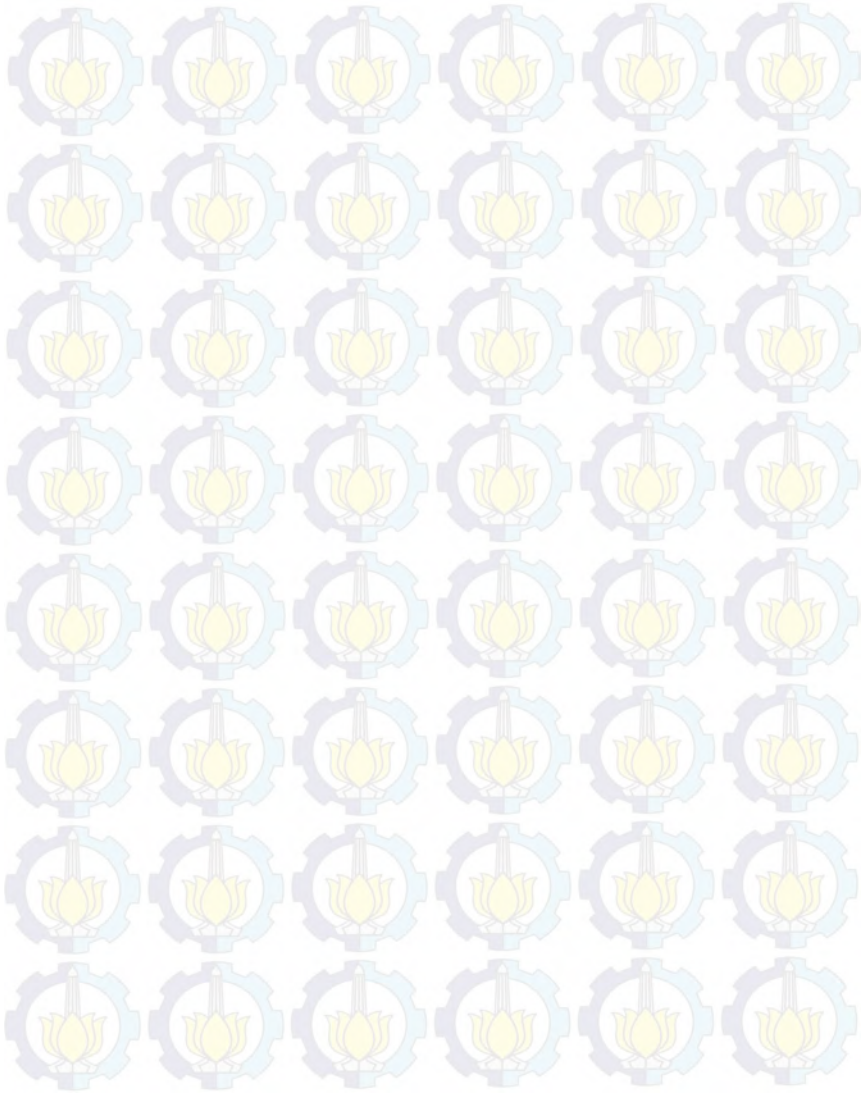


TABLE OF CONTENT

AUTHENTICITY	iii
APPROVAL SHEET	v
ABSTRAK	vii
ABSTRACT	ix
PREFACE	xi
TABLE OF CONTENT	xiii
ILLUSTRATION	xvi
TABLES	xix
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Problems	2
1.3 Problems Constraints	2
1.4 Objectives	3
1.5 Methodology	3
1.6 Systematics	4
1.7 Relevances	5
CHAPTER 2 COUPLED TANKS CONTROL SYSTEM	7
2.1 Control System	7
2.2 Control System Configuration	8
2.2.1 Single loop	8
2.2.2 Cascade	9
2.3 Coupled Tanks	11
2.4 Modelling System Coupled Tanks	12
2.4.1 Pump Modelling	13
2.4.2 Pipe Modelling	13
2.4.3 Tank Modelling	14
2.4.4 Sensor Level Modelling	18
2.5 Flow Plant Coupled Tanks Modelling	20
2.5.1 Sensor Flow Modelling	20
2.6 PID Controller	20
2.7 Disturbances	22
2.7.1 Load Disturbances	22
2.7.2 Measurement Noise	22
2.8 Artificial Neural Network	23
2.8.1 Basic Concept Neural Network Modelling	23
2.8.2 Learning in Neural Network	24

2.9	PID – Neural Network	24
CHAPTER 3 SYSTEM DESIGN		27
3.1	System Design Preview	27
3.2	System Coupled Tanks Modelling	28
3.2.1	Pump Gain Transfer Function.....	28
3.2.2	Pipe Transfer Function.....	28
3.2.3	Tank Transfer Function.....	28
3.2.4	Sensor Level Transfer Function	29
3.2.5	Transfer function All.....	29
3.2.6	Flow Modelling	29
3.3	Design Controller	30
3.3.1	Design PID Controller using Root locus.....	30
3.3.2	Design PID Controller using Miluse Viteckova Method	33
3.3.3	Design PID-Neural Network Controller	34
3.4	Design Human Machine Interface (HMI)	39
CHAPTER 4 EXPERIMENT and ANALYSIS		43
4.1	System Simulation without Controller.....	43
4.1.1	Closed Loop Level Process Respon	43
4.1.2	Open Loop Flow Process	44
4.2	System Simulation using PID Controller	44
4.2.1	Simulation Level Control Single Loop Set point Variation	45
4.2.2	Simulation Level Control Single Loop Controller Parameter Variation (K_p , τ_i , dan τ_d)	46
4.2.3	Simulation Level Control Single Loop with Disturbance	49
4.2.4	Simulation Level Control Single Loop Noise variation.....	50
4.2.5	Simulation Flow Control System	51
4.2.6	Simulation Water Level Control System with Cascade Configuration	51
4.2.7	Simulation Water Level Control System with Cascade Configuration with noise.....	52
4.3	System Simulation using PID Neural-Network Controller	54
4.3.1	Simulation Water Level Control System	55
4.3.2	Simulation Using Learning Rate Variation	56
4.3.3	System Simulation with Disturbances	57
BAB 5 CONCLUSION and SUGGESTION		61
5.1	Conclusion	61
5.2	Suggestion.....	61
BIBLIOGRAPHY		63
ENCLOSURE		65

DAFTAR GAMBAR

Gambar 2.1	Diagram blok sistem pengaturan	7
Gambar 2.2	Sistem pengaturan <i>level single loop</i>	8
Gambar 2.3	Diagram blok sistem pengaturan <i>level single loop</i>	9
Gambar 2.4	Diagram blok sistem pengaturan konfigurasi <i>cascade</i>	9
Gambar 2.5	Sistem pengaturan <i>level</i> air dengan konfigurasi <i>cascade</i>	10
Gambar 2.6	Model <i>plant coupled tank</i>	11
Gambar 2.7	Diagram blok pemodelan sistem pengaturan <i>level</i>	12
Gambar 2.8	Alur pemodelan <i>flow</i>	19
Gambar 2.9	Model <i>sump tank</i>	19
Gambar 2.10	Diagram blok kontroler PID	21
Gambar 2.11	Diagram blok sistem pengaturan dengan <i>disturbance</i>	22
Gambar 2.12	Model neuron	23
Gambar 2.13	Diagram blok sistem pengaturan dengan kontroler PID <i>Neural Network</i>	25
Gambar 3.1	Diagram blok sistem pengaturan <i>level plant coupled tank</i>	27
Gambar 3.2	<i>Root locus</i> dari sistem G_2	32
Gambar 3.3	<i>Root locus plant</i> setelah penambahan kontroler PID	32
Gambar 3.4	Struktur <i>Neural Network</i> untuk <i>autotuning</i> PID	35
Gambar 3.5	<i>Image Navigator</i> pada <i>DSC Module</i>	40
Gambar 3.6	HMI dari <i>plant coupled tank</i>	41
Gambar 3.7	Program simulasi pengaturan <i>level</i> dengan menggunakan <i>Control and Simulation Module</i>	42
Gambar 4.1	Respon closed loop sistem	43
Gambar 4.2	Respon <i>open loop</i> proses <i>flow</i>	44
Gambar 4.3	Respon <i>plant</i> dengan perubahan <i>setpoint</i>	45
Gambar 4.4	Respon <i>plant</i> dengan variasi nilai K_p untuk sistem G_2	46
Gambar 4.5	Respon <i>plant</i> dengan variasi nilai τ_i untuk sistem G_2	47
Gambar 4.6	Respon <i>plant</i> dengan variasi nilai τ_d untuk sistem G_2	48
Gambar 4.7	Respon <i>plant</i> dengan pemberian beban	49
Gambar 4.8	Respon <i>plant</i> dengan diberi <i>noise</i>	50
Gambar 4.9	Respon proses <i>flow</i> setelah diberi kontroler	51
Gambar 4.10	Respon pengaturan <i>level</i> konfigurasi <i>cascade</i> dibandingkan dengan konfigurasi <i>single loop</i>	52
Gambar 4.11	Diagram blok sistem pengaturan konfigurasi <i>cascade</i> dengan <i>disturbance</i>	53

Gambar 4.12	Respon dengan <i>disturbance</i> pada pengaturan <i>level</i> dengan konfigurasi <i>cascade</i> dan konfigurasi <i>single loop</i>	53
Gambar 4.13	Efek pemberian beban pada respon <i>plant</i>	54
Gambar 4.14	Respon <i>plant</i> sistem pengaturan <i>level</i> dengan kontroler PID <i>Neural Network</i>	55
Gambar 4.15	Perubahan parameter kontroler keluaran <i>Neural Network</i>	56
Gambar 4.16	Respon <i>plant</i> dengan variasi <i>learning rate</i>	57
Gambar 4.17	Simulasi sistem pengaturan <i>level</i> dengan pemberian beban untuk konfigurasi <i>single loop</i> dengan kontroler PID-NN.....	57
Gambar 4.18	Simulasi sistem pengaturan <i>level</i> dengan pemberian beban untuk konfigurasi <i>cascade</i> dengan kontroler PID-NN.....	58
Gambar 4.19	Perubahan parameter kontroler keluaran <i>Neural Network</i> untuk PID <i>Neural Network</i> dengan konfigurasi <i>single loop</i>	59
Gambar 4.20	Perubahan parameter kontroler keluaran <i>Neural Network</i> untuk PID <i>Neural Network</i> dengan konfigurasi <i>cascade</i>	59

DAFTAR TABEL

Tabel 2.1	Paramater pemodelan <i>plant</i>	13
Tabel 3.1	Nilai Parameter R dan C pemodelan tangki.....	28
Tabel 3.2	<i>Tuning</i> Parameter Kontroler PID dengan metode Miluse Viteckova.....	33
Tabel 3.3	Inisiasi <i>weight input layer</i> menuju <i>hidden layer</i>	39
Tabel 3.4	Inisiasi <i>weight hidden layer</i> menuju <i>output layer</i>	39



Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Pada Bab ini berisikan penjelasan awal mengenai penelitian yang akan dilakukan. Penjelasan tersebut meliputi latar belakang masalah, permasalahan yang diangkat, batasan masalah yang digunakan, tujuan dari penelitian ini, sistematika penulisan, penyusunan laporan dan juga relevansi.

1.1 Latar Belakang

Pada industri proses seperti industri *petro-chemical*, pembuatan kertas, dan industri pengolahan air yang melibatkan cairan, suatu cairan akan dipompa dan dialirkan dari satu tangki ke tangki yang lain untuk diolah. Ada beberapa jenis tangki yang banyak digunakan pada industri proses. Ada jenis tangki yang tidak terhubung dengan tangki yang lain (*single tank*) dan ada yang tersusun dari beberapa tangki yang saling berhubungan (*coupled tank*) [1].

Coupled tank digolongkan pada *plant* dengan sistem *Multi Input Multi Output* (MIMO) atau proses *multivariable*. Pengaturan pada *plant* MIMO lebih sulit dibanding *plant* dengan sistem *Single Input Single Output* (SISO). Karena adanya interaksi silang antara variabel masukan dan keluaran, sehingga metode kontrol yang diterapkan pada sistem SISO bisa tidak dapat digunakan secara baik pada sistem MIMO [2].

Pada pengaturan *level* air di *coupled tank*, *level* pada tangki kedua akan dipengaruhi oleh aliran cairan dari tangki pertama. Pemindahan cairan dari satu tangki ke tangki yang lain menyebabkan berubahnya *level* cairan dalam tangki. Dalam pengaturan *level*, pemindahan cairan biasa disebut sebagai pembebanan pada *level*. Perubahan beban ini dapat mempengaruhi dari kinerja kontroler.

Kontroler yang umum digunakan pada industri adalah kontroler PID karena kesederhanaan strukturnya dan kehandalannya [3]. Pada setiap perangkat kontrol di industri seperti *programmable logic controller* (PLC) tersedia kontroler PID dengan tambahan fitur *autotuning* untuk memudahkan operator menjalankan sistem kontrol umpan balik dengan PID.

Tuning parameter kontroler PID dilakukan dengan menyesuaikan nilai *proportional gain* (K_p), *integral time* (T_i), *derivative time* (T_d) agar tercapai spesifikasi yang diinginkan. Pada prakteknya di dunia industri *tuning* parameter PID sering dilakukan dengan cara *trial and error*.

Prosedur ini akan memakan banyak waktu dan membutuhkan seorang operator yang handal [4]. Akan tetapi apabila *plant* sering mengalami perubahan beban maka kontroler PID perlu dilakukan *tuning* parameter ulang agar tetap memenuhi spesifikasi kontrol yang diharapkan. Sehingga perlu diterapkan metode *autotuning* parameter kontroler PID yang dapat meminimumkan akan adanya *error* bila *tuning* dilakukan dengan prosedur *trial and error* dan tetap menjaga spesifikasi kontrol yang diharapkan.

Metode *autotuning* yang akan digunakan adalah penggunaan algoritma cerdas *Neural Network*. Algoritma *Neural Network* ini digunakan untuk mempelajari perubahan beban pada *plant* dan akan melakukan *tuning* parameter kontroler PID secara otomatis selama proses berlangsung sehingga dapat memenuhi spesifikasi kontrol yang diinginkan.

1.2 Perumusan Masalah

Permasalahan yang sering muncul adalah perubahan parameter *plant* akan menyebabkan performa kontroler PID tidak sesuai dengan kondisi parameter mula-mula. Untuk memenuhi spesifikasi kontroler yang diinginkan pada setiap nilai parameter *plant*, maka parameter kontroler PID harus dilakukan *tuning* ulang setiap kali terjadi perubahan *plant*. Pada prakteknya *tuning* parameter kontroler PID sering dilakukan dengan cara *trial and error*, akan tetapi cara tersebut memiliki resiko cukup tinggi yang dapat berdampak pada *plant* apabila *error* dalam melakukan *tuning* parameter, dan juga prosedur ini akan memakan banyak waktu dan membutuhkan operator yang handal atau telah berpengalaman. Sehingga diperlukan suatu metode *autotuning* parameter kontroler PID yang dapat memperbarui parameter kontroler PID untuk tetap menjaga spesifikasi kontrol sesuai dengan yang diharapkan

1.3 Batasan Masalah

Plant coupled tank merupakan *plant* dengan dua tangki atau lebih yang bisa digolongkan pada plant MIMO, SISO atau SIMO. Pada tugas akhir ini yang akan dibahas adalah mengenai pengaturan level dengan Sistem SISO. Implementasi kontroler pada simulator *plant* yang dibuat pada LabVIEW, belum hingga ke implementasi pada *plant* real. Pengendalian *level* air diterapkan pada tangki dua dari *plant coupled tank*. Dalam pengujian *noise* pada sistem pengaturan *level* coupled tank ini digunakan *Additive White Gaussian Noise (AWGN)*.

1.4 Tujuan Penelitian

Penerapan metode *autotuning* kontroler PID menggunakan algoritma cerdas *Neural Network* untuk sistem pengaturan *level* pada simulator *plant coupled tank* dan membandingkan unjuk kerjanya dengan kontroler PID konvensional.

1.5 Metodologi

Dalam melaksanakan penelitian ini, dilakukan melalui beberapa tahapan yang dilalui diantaranya:

1. Studi Literatur

Studi literatur ini dilakukan untuk mengenal topik penelitian dengan baik. Studi literatur ini meliputi mencari dan mempelajari literatur, buku-buku, dan *paper* yang relevan dengan penelitian ini mengenai materi sistem pengaturan pada *plant coupled tank*, sistem pengaturan *cascade*, jaringan syaraf tiruan (*Artificial Neural Network*), kontroler PID, dan PID – *Neural Network*.

2. Pemodelan *Plant*

Pada penelitian ini *plant* yang digunakan adalah simulator *plant coupled tank* yang dibangun pada perangkat lunak LabVIEW, sehingga dibutuhkan model untuk dapat memberikan gambaran dari *real system*. Pemodelan *plant* ini dilakukan dengan menurunkan persamaan matematis dari *plant coupled tank*.

3. Perancangan Kontroler

Berdasarkan model *plant* yang telah didapat, kemudian model tersebut diuji tanpa kontroler. Dari permasalahan yang muncul dari hasil pengujian tanpa kontroler tersebut maka akan dirancang kontroler sesuai dengan spesifikasi respon yang diinginkan. Metode kontroler yang akan digunakan adalah kontroler PID dengan perhitungan gain (K_p , τ_i , dan τ_d) menggunakan metode *root locus* dan PID – *Neural Network* dengan algoritma pembelajaran *Backpropagation*.

4. Simulasi

Hasil pemodelan sistem dan perancangan kontroler disimulasikan terlebih dahulu sebelum diimplementasikan dengan menggunakan bantuan perangkat lunak simulasi.

5. Implementasi

Hasil simulasi yang didapat kemudian diimplementasikan pada simulator *plant coupled tank*. Hasil implementasi ini kemudian

dianalisa sesuai dengan teori yang digunakan. Dari hasil analisa tersebut kemudian bisa didapatkan mengenai kesimpulan dari Tugas Akhir ini

6. Penulisan Buku Tugas Akhir

Penulisan buku Tugas Akhir ini dilakukan sebagai dokumentasi dari hasil penelitian yang dilakukan. Penulisan buku ini berupa laporan ilmiah yang mencakup semua proses pengerjaan penelitian ini

1.6 Sistematika Penulisan

Secara keseluruhan, sistematika penulisan Tugas Akhir ini dibagi menjadi lima bab dengan sistematika sebagai berikut:

BAB 1 : PENDAHULUAN

Bab ini meliputi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, sistematika laporan, dan relevansi penulisan Tugas Akhir

BAB 2 : SISTEM PENGATURAN *PLANT COUPLED TANK*

Teori dasar yang digunakan sebagai pendukung dalam penelitian ini akan dibahas pada BAB 2 ini diantaranya *plant coupled tank*, kontroler PID, dan PID –*Neural Network*

BAB 3 : PERANCANGAN SISTEM

Tahapan-tahapan dalam perancangan sistem yang meliputi desain dan perancangan kontroler PID dan *autotuning* PID dengan *Neural Network* berdasarkan teori pada BAB 2. Selain itu dibahas juga mengenai tahapan algoritma kontrol yang dibuat.

BAB 4 : HASIL PENGUJIAN DAN ANALISA

Pelaporan data hasil simulasi disertai dengan analisis mengenai hasil yang didapatkan dari pengujian sistem yang dilakukan

BAB 5 : KESIMPULAN DAN SARAN

Penulisan kesimpulan berdasarkan data yang telah diambil dan dianalisa. Selain itu juga disertakan saran mengenai kekurangan dari penelitian ini dan rencana riset selanjutnya yang bisa dikembangkan.

1.7 Relevansi

Manfaat yang didapat dari penelitian ini nantinya diharapkan kotroler PID dengan *autotuning* menggunakan *Neural Network* mampu diterapkan pada *plant coupled tank* atau pengaturan proses lainnya dan mampu mengatasi permasalahan khususnya perubahan beban pada *plant*.

BAB 2

SISTEM PENGATURAN *PLANT COUPLED TANK*

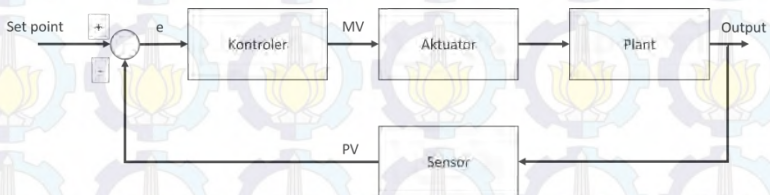
Kegiatan penelitian dan perancangan merupakan hasil mengkaji teori-teori yang sudah ada sebelumnya. Seluruh teori yang digunakan kemudian disajikan untuk memperkuat argumen penulis dalam penelitian ini.

Pada Bab ini, berisi tentang teori penunjang dari berbagai pustaka atau literatur seperti buku dan *paper* yang mendukung penelitian ini. Teori penunjang pada Bab ini meliputi *plant coupled tank*, *Neural Network*, kontroler PID, dan kontroler PID-*Neural Network*.

2.1 Sistem Pengaturan

Sistem merupakan sekumpulan perangkat yang saling bekerja sama untuk mencapai tujuan tertentu. Pengaturan adalah sebuah metode untuk membuat nilai suatu variabel mencapai nilai tertentu yang kita inginkan. Contohnya menjaga ketinggian *level* air pada tangki pada ketinggian 50 cm. Secara umum, maka sistem pengaturan adalah sekumpulan perangkat atau elemen yang saling bekerja sama untuk mencapai objektif pengaturan atau kontrol sesuai dengan yang kita kehendaki.

Sistem pengaturan terdiri dari beberapa elemen yang dapat dijelaskan lebih detail menggunakan diagram blok seperti pada Gambar 2.1.



Gambar 2.1 Diagram blok sistem pengaturan

Elemen utama pada sistem pengaturan sesuai dengan diagram blok pada Gambar 2.1 adalah *plant*, kontroler, aktuator, dan sensor. *Plant* merupakan elemen yang dikendalikan nilai keluarannya. Kontroler adalah yang bertugas mengendalikan keluaran dari *plant* dan memperbaiki performa dari suatu sistem pengaturan. Aktuator atau biasa disebut *final*

control element adalah perangkat yang berhubungan langsung dengan *plant*. Sensor berfungsi untuk membaca nilai keluaran dari *plant*.

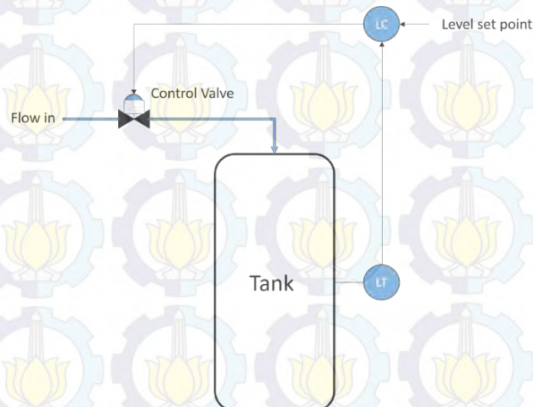
Pada sistem pengaturan ada tiga parameter nilai yaitu *set point* atau *set value* (SV), *manipulated value* (MV), dan *process value* (PV). *Set point* merupakan nilai dari keluaran proses atau *plant* yang kita kehendaki. *Manipulated value* (MV) atau biasa disebut dengan sinyal kontrol merupakan keluaran dari kontroler, sedangkan *process value* (PV) merupakan hasil pembacaan dari sensor terhadap nilai keluaran dari *plant*.

2.2 Konfigurasi Sistem Pengaturan

Sistem pengaturan yang ada pada Gambar 2.1 digambarkan sebagai *closed loop system* dikarenakan adanya umpan balik dari *plant* menuju kontroler. Keputusan pemilihan konfigurasi ini bertujuan untuk objektif kontrol dapat terpenuhi sesuai dengan spesifikasi yang dikehendaki. Konfigurasi *closed loop system* yang sering dijumpai adalah *single loop* dan *cascade*.

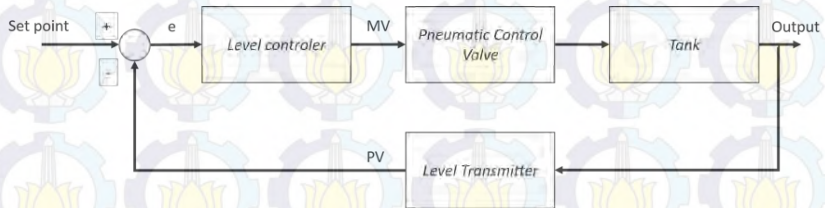
2.2.1 Single Loop

Konfigurasi *single loop* merupakan konfigurasi sistem kendali paling sederhana dimana hanya terdapat satu buah *loop* dengan satu kontroler untuk memenuhi objektif kontrol. Diagram blok dari sistem pengaturan dengan konfigurasi *single loop* ini sama seperti pada Gambar 2.1. Contoh implementasi dari konfigurasi ini seperti pada Gambar 2.2



Gambar 2.2 Sistem pengaturan level single loop

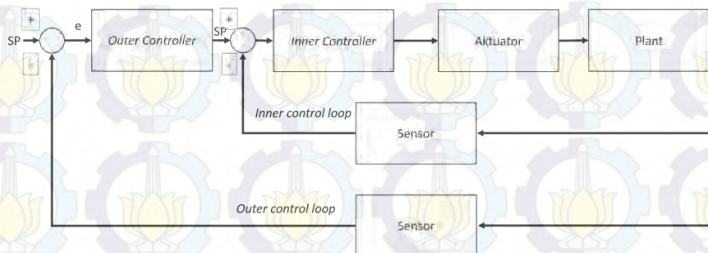
Sesuai dengan diagram blok pada Gambar 2.1 yang bertindak sebagai *plant* adalah tangki dengan proses yang dikendalikan adalah ketinggian air pada tangki, sebagai kontroler adalah *level controller*, *pneumatic control valve* berfungsi sebagai aktuatur atau *final control element*, dan *level transmitter* berfungsi sebagai sensor. Apabila direpresentasikan dalam diagram blok maka seperti pada Gambar 2.3.



Gambar 2.3 Diagram blok sistem pengaturan *level single loop*

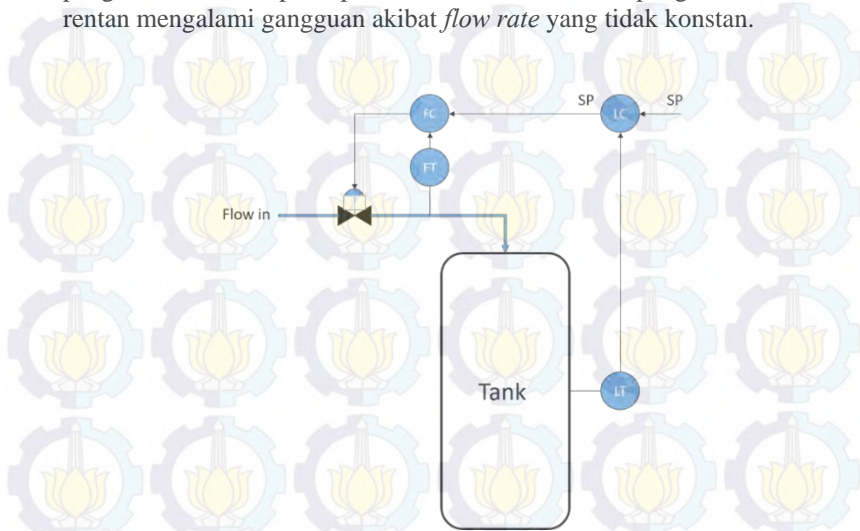
2.2.2 Cascade

Struktur sistem pengaturan *cascade* atau biasa disebut juga dengan *master-slave controller*, terdiri atas dua atau lebih *loop* kontrol yang saling berinteraksi dengan diagram blok seperti pada Gambar 2.4. Proses yang berjalan lebih cepat dijadikan sebagai *inner loop*, sedangkan untuk proses yang berjalan lebih lambat sebagai *outer loop*. Dalam sistem pengaturan konfigurasi *cascade* ini terdapat dua kontroler yang digunakan yaitu *inner controller* dan *outer controller*, dimana *outer controller* memberikan *set point* pada *inner controller*, dua buah sensor atau *measurement device* untuk mengukur dua variabel proses yang berbeda, dan hanya satu aktuatur atau *final control element* yang digunakan.



Gambar 2.4 Diagram blok sistem pengaturan konfigurasi *cascade*

Salah satu contoh implementasi dari konfigurasi ini adalah pengaturan *level* seperti pada Gambar 2.5. Proses pengendalian *level* rentan mengalami gangguan akibat *flow rate* yang tidak konstan.



Gambar 2.5 Sistem pengaturan *level* air dengan konfigurasi *cascade*

Pada proses pengendalian *level* air ini yang bertindak sebagai *inner loop* adalah proses *flow* karena memiliki respon yang lebih cepat daripada proses *level*, sehingga proses *level* ditempatkan pada *outer loop*. Kontroler dari proses *level* ini akan memberikan *set point* kepada kontroler proses *flow* yang ditempatkan pada *inner loop*. Dari Gambar 2.5 dapat dilihat bahwa aktuatur atau *final control element* yang digunakan hanya satu yaitu *pneumatic control valve* untuk mengatur laju aliran air yang masuk ke tangki. Untuk sensor digunakan dua jenis yang berbeda yaitu *level transmitter* yang ditempatkan pada *outer loop* dan *flow transmitter* yang ditempatkan pada *inner loop*.

Konfigurasi *cascade* ini memiliki beberapa kelebihan dibanding dengan *single loop* yaitu:

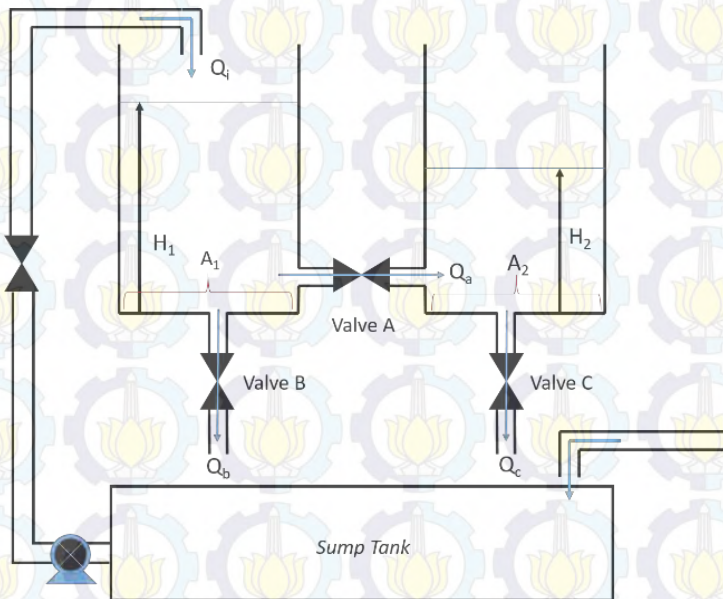
- 1) Mampu meredam *disturbance* yang terjadi pada *inner loop* sehingga tidak berakibat pada *outer loop*
- 2) Mampu mengkompensasi efek non-linear aktuatur pada *inner loop* sehingga bisa didapatkan respon yang linear pada *outer loop*

Pada konfigurasi *cascade* ini juga memiliki beberapa kelemahan dibanding dengan konfigurasi *single loop* yaitu:

- 1) Membutuhkan tambahan sensor atau *measurement device* untuk bekerja
- 2) Adanya kontroler tambahan yang perlu di *tuning* parameter kontrolernya
- 3) Strategi kontrol yang diterapkan lebih kompleks.

2.3 Coupled Tank

Plant coupled tank merupakan tangki dengan konfigurasi dua tangki atau lebih yang saling berhubungan dengan sebuah pipa atau saluran air seperti pada Gambar 2.6. Adanya hubungan antara tangki ini membuat *level* cairan pada setiap tangki saling berinteraksi atau berhubungan. Konfigurasi tangki seperti ini banyak digunakan pada industri proses seperti industri *petro-chemical*, pembuatan kertas, dan industri pengolahan air.



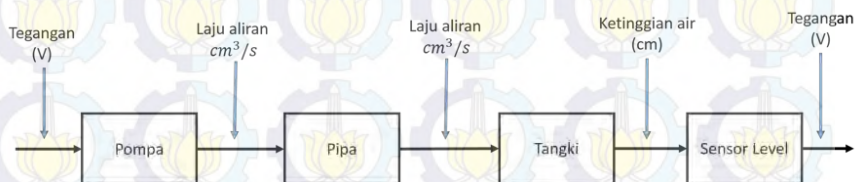
Gambar 2.6 Model *plant coupled tank*

Level cairan pada tangki pertama ditunjukkan sebagai H_1 dan H_2 untuk *level* tangki kedua. *Flow* cairan yang masuk pada tangki dilambangkan sebagai Q_i . Untuk *flow* cairan yang keluar dari tangki ditunjukkan sebagai Q_b untuk tangki pertama, Q_c tangki kedua, dan Q_a sebagai *flow* interaksi antara tangki pertama dan tangki kedua. Sistem *coupled tank* dapat dikonfigurasi sebagai sistem *Single Input Single Output* (SISO), *Multi Input Multi Output* (MIMO), atau *Single Input Multi Output* (SIMO) berdasarkan manipulasi masukan pada pompa dan daerah kerja dari *rotary valve* yang terdapat pada *plant coupled tank*. Jika berdasar model pada Gambar 2.1 maka *plant* ini bisa digolongkan sebagai sistem SIMO karena hanya memiliki satu masukan berupa aliran masuk air pada tangki satu dan untuk keluaran adalah *level* pada kedua tangki.

2.4 Pemodelan Sistem Pengaturan Level Plant Coupled Tank

Sistem yang akan dirancang adalah sistem pengaturan *level* dengan konfigurasi SISO yaitu pengaturan *level* pada tangki ke dua, dengan ditunjukkan masukan berupa tegangan pada pompa air untuk memberi aliran air masuk ke tangki dan keluaran berupa *level* air pada tangki satu atau dua. Model dari *plant coupled tank* seperti pada Gambar 2.6.

Proses pemodelan dimulai dari pemodelan pompa, pipa, tangki pada *plant coupled tank*, dan sensor yang digunakan seperti pada Gambar 2.7.



Gambar 2.7 Diagram blok pemodelan sistem pengaturan level

Parameter yang digunakan pada pemodelan sistem pengaturan *level* ini dapat dilihat pada Tabel 2.1. Dalam pemodelan suatu sistem diperlukan juga untuk mengetahui karakteristik dari komponen setiap penyusun sistem. Sebagai contoh pada pompa perlu diketahui berapa tegangan maksimal kerja pompa, cara kerja dari pompa. Dengan diketahui dengan detail mengenai karakteristik dari sistem maka model matematis yang akan dibuat bisa lebih akurat dan juga dalam *real* sistem maka akan

diketahui permasalahan yang timbul apabila ada karakteristik dari setiap komponen yang diabaikan.

Tabel 2.1 Paramater pemodelan *plant*

Simbol	Keterangan	Satuan
Q_i	Debit air yang masuk ke tangki	cm^3 / s
Q_a, Q_c	Debit air yang keluar dari tangki 1 dan 2	cm^3 / s
Q_b	Debit air yang keluar dari tangki 1 menuju tangki 2	cm^3 / s
H_1, H_2	Ketinggian air dalam tangki 1 dan 2	cm
A_1, A_2	Luas alas tangki	cm^2
a	Diameter pipa	cm
R	Resistansi control <i>valve</i>	s / cm^2
L	Panjang pipa	cm
C	Kapasitansi tangki	cm^2

2.4.1 Pemodelan Pompa

Pompa yang digunakan dalam sistem ini adalah pompa DC dengan rentang masukan tegangan 0 – 10 V dan debit air keluar dengan rentang 0 – 3.5 l/menit. Dari pemodelan ini akan didapatkan nilai penguatan yang dilambangkan dengan K_1 . K_1 dirumuskan dengan debit air keluar dari pompa terhadap masukan berupa tegangan seperti pada Persamaan (2.1)

$$K_1 = \frac{\text{Rentang debit air yang keluar dari pompa (cm}^3/s\text{)}}{\text{Rentang tegangan masukan (V)}} \quad (2.1)$$

2.4.2 Pemodelan Pipa

Pemodelan pipa pada sistem pengaturan *level* ini untuk mencari tahu besarnya waktu tunda yang dilambangkan dengan θ yang diakibatkan volume pipa yang digunakan. Semakin besar volume pipa maka aliran air dari pompa menuju ke tangki akan semakin lama sehingga waktu tunda dalam sistem juga semakin besar. Hubungan waktu tunda dengan panjang pipa dapat dirumuskan pada Persamaan (2.2)

$$\theta = \frac{\text{Volume pipa (cm}^3\text{)}}{\text{Debit air yang masuk ke pipa (cm}^3\text{/s)}} \quad (2.2)$$

2.4.3 Pemodelan Tangki

Konfigurasi tangki pada *plant coupled tank* ini merupakan tangki yang saling berhubungan. Parameter utama dalam sistem tangki air adalah kapasitansi tangki dan resistansi jalur pembuangan. Kapasitansi tangki ini mirip seperti pada kapasitor. Yaitu ketika proses pengisian air pada tangki mirip dengan proses *charge*, dan ketika pembuangan air seperti proses *discharge* pada kapasitor. Resistansi ini didefinisikan sebagai hambatan pada saat proses pembuangan air dikarenakan debit air yang keluar dipengaruhi oleh bukaan dari *valve*.

Pemodelan matematis pada tangki ini berdasarkan hukum kesetimbangan massa seperti pada Persamaan (2.3)

$$Q = Q_{in} - Q_{out} \quad (2.3)$$

Berdasarkan Persamaan (2.3) maka perubahan *level* air pada tangki 1 dan tangki 2 di *plant coupled tank* seperti pada Persamaan (2.4) dan Persamaan (2.6).

$$C_1 \frac{dh_1}{dt} = Q_i - Q_a - Q_b \quad (2.4)$$

$$C_2 \frac{dh_2}{dt} = Q_a - Q_c \quad (2.5)$$

Dimana C_1, C_2 adalah efek kapasitansi pada tangki. Kapasitansi suatu tangki sama dengan luas permukaan penampang tangki. Tangki yang dimodelkan ini memiliki bentuk alas yaitu persegi dengan ukuran sisinya 17 cm. Sehingga didapat parameter untuk C_1, C_2 seperti pada Persamaan (2.6) . Dimana A merupakan luas penampang dari tangki.

$$C = \frac{\text{Perubahan volume air (cm}^3\text{)}}{\text{Perubahan ketinggian air (cm)}} = A = s^2 \quad (2.6)$$

Untuk R_1, R_2 dan R_3 didefinisikan sebagai resistansi atau hambatan saat terjadi proses pembuangan air melalui *control valve*. Hal ini dikarenakan debit air yang keluar bergantung pada bukaan dari *control valve*. Persamaan untuk R_1, R_2 dan R_3 ini seperti pada Persamaan (2.7)

$$R = \frac{\text{Perubahan ketinggian air (cm}^3\text{)}}{\text{Perubahan debit air yang keluar (cm}^3\text{/s)}} \quad (2.7)$$

Dari Persamaan (2.7) dapat diketahui debit air yang keluar seperti pada Persamaan (2.8)

$$Q_0 = \frac{h_0}{R} \quad (2.8)$$

Dari Persamaan (2.8) didapat persamaan untuk masing-masing debit air yang keluar dari setiap tangki seperti pada Persamaan (2.9), (2.10), dan (2.11)

$$Q_a = \frac{h_1 - h_2}{R_1} \quad (2.9)$$

$$Q_c = \frac{h_2}{R_2} \quad (2.10)$$

$$Q_b = \frac{h_1}{R_3} \quad (2.11)$$

Kemudian disubstitusikan Persamaan (2.9), (2.10) dan (2.11) pada Persamaan (2.4) dan (2.5) sehingga didapat Persamaan seperti pada Persamaan (2.12) dan (2.13)

$$C_1 \frac{dh_1}{dt} + \frac{h_1}{R_1} + \frac{h_1}{R_3} = Q_i + \frac{h_2}{R_1} \quad (2.12)$$

$$C_2 \frac{dh_2}{dt} + \frac{h_2}{R_1} + \frac{h_2}{R_2} = \frac{h_1}{R_1} \quad (2.13)$$

Dari Persamaan (2.12) dan (2.13) kemudian dilakukan Transformasi *Laplace* sehingga diperoleh Persamaan (2.14) dan (2.15)

$$\left(C_1 s + \frac{1}{R_1} + \frac{1}{R_3} \right) H_1(s) = Q_i(s) + \frac{1}{R_1} H_2(s) \quad (2.14)$$

$$\left(C_2 s + \frac{1}{R_1} + \frac{1}{R_2} \right) H_2(s) = \frac{1}{R_1} H_1(s) \quad (2.15)$$

Dari Persamaan (2.14) diperoleh persamaan untuk $H_1(s)$ seperti pada Persamaan (2.16)

$$H_1(s) = \frac{Q_i(s) + \frac{1}{R_1} H_2(s)}{C_1 s + \frac{1}{R_1} + \frac{1}{R_3}} \quad (2.16)$$

Dari Persamaan (2.16) disubstitusikan ke Persamaan (2.15) sehingga diperoleh Persamaan (2.17)

$$\left(C_2 s + \frac{1}{R_1} + \frac{1}{R_2} \right) H_2(s) = \frac{1}{R_1} \left(\frac{Q_i(s) + \frac{1}{R_1} H_2(s)}{C_1 s + \frac{1}{R_1} + \frac{1}{R_3}} \right) \quad (2.17)$$

Kemudian dilakukan perhitungan dari Persamaan (2.17) didapatkan Persamaan (2.18), (2.19), dan (2.20)

$$\left(R_1 C_2 s + 1 + \frac{R_1}{R_2} \right) H_2(s) = \frac{Q_i(s) + \frac{1}{R_1} H_2(s)}{C_1 s + \frac{1}{R_1} + \frac{1}{R_3}} \quad (2.18)$$

$$\left(C_1 s + \frac{1}{R_1} + \frac{1}{R_3} \right) \left(R_1 C_2 s + 1 + \frac{R_1}{R_2} \right) H_2(s) = Q_i(s) + \frac{1}{R_1} H_2(s) \quad (2.19)$$

$$\left(R_1 C_1 C_2 s^2 + C_1 s + \frac{R_1}{R_2} C_1 s + C_2 s + \frac{R_1}{R_3} C_2 s + \frac{1}{R_3} + \frac{1}{R_2} + \frac{R_1}{R_2 R_3} \right) H_2(s) = Q_i(s) \quad (2.20)$$

Dari Persamaan (2.20) diperoleh persamaan fungsi alih antara $H_2(s)$ dan $Q_i(s)$ seperti pada Persamaan (2.21)

$$\frac{H_2(s)}{Q_i(s)} = \frac{1}{R_1 C_1 C_2 s^2 + \left(C_1 + \frac{R_1}{R_2} C_1 + C_2 + \frac{R_1}{R_3} C_2 \right) s + \frac{1}{R_3} + \frac{1}{R_2} + \frac{R_1}{R_2 R_3}} \quad (2.21)$$

Untuk pemodelan *level* air pada tangki 1, maka diperoleh Persamaan untuk $H_2(s)$ seperti pada Persamaan (2.22)

$$H_2(s) = \left(R_1 C_1 s + 1 + \frac{R_1}{R_3} \right) H_1(s) - R_1 Q_i(s) \quad (2.22)$$

Dari Persamaan (2.22) ini kemudian disubstitusi ke Persamaan (2.15) sehingga didapatkan Persamaan (2.23). Dengan melakukan perhitungan pada Persamaan (2.23) maka bisa didapatkan Persamaan (2.24) mengenai hubungan antara H_1 yaitu *lever* air pada tangki satu dengan Q_i , aliran masuk air ke tangki satu.

$$\left(C_2 s + \frac{1}{R_1} + \frac{1}{R_2} \right) \left(\left(R_1 C_1 s + 1 + \frac{R_1}{R_3} \right) H_1(s) - R_1 Q_i(s) \right) = \frac{1}{R_1} H_1(s) \quad (2.23)$$

$$\left(R_1 C_1 C_2 s^2 + C_1 s + \frac{R_1}{R_2} C_1 s + C_2 s + \frac{R_1}{R_3} C_2 s + \frac{1}{R_2} + \frac{1}{R_3} + \frac{R_1}{R_2 R_3} \right) H_1(s) = \left(R_1 C_2 s + 1 + \frac{R_1}{R_2} \right) Q_i(s) \quad (2.24)$$

Didapat fungsi alih untuk pemodelan *level* air tangki 1 seperti pada Persamaan (2.25)

$$\frac{H_1(s)}{Q_i(s)} = \frac{R_1 C_2 s + \left(1 + \frac{R_1}{R_2} \right)}{R_1 C_1 C_2 s^2 + \left(C_1 + \frac{R_1}{R_2} C_1 + C_2 + \frac{R_1}{R_3} C_2 \right) s + \left(\frac{1}{R_2} + \frac{1}{R_3} + \frac{R_1}{R_2 R_3} \right)} \quad (2.25)$$

2.4.4 Pemodelan Sensor *Level*

Sensor *level* yang akan dimodelkan ini bertipe *floating* dengan keluaran dari sensor berupa tegangan dengan rentang 0 – 10 V dan rentang ketinggian antara 0-18 cm. Dari pemodelan ini akan didapat nilai *gain* / penguatan K_2 . K_2 dirumuskan berupa perbandingan rentang kerja dari pompa terhadap masukan berupa tegangan seperti pada Persamaan (2.26)

$$K_2 = \frac{\text{Rentang tegangan keluaran (V)}}{\text{Rentang ketinggian air dalam tangki (cm)}} \quad (2.26)$$

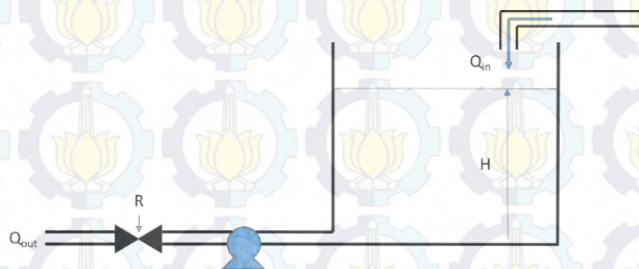
2.5 Pemodelan Sistem Pengaturan *Flow Plant Coupled Tank*

Konfigurasi kontrol yang digunakan pada penelitian ini adalah konfigurasi *cascade* antara proses *level* dan *flow*. Proses *flow* ini dimodelkan sesuai dengan alur seperti pada Gambar 2.8. Proses pemodelan dimulai dari pemodelan *flow* yang keluar dari tangki *sump tank*, pemodelan pompa, pipa, dan sensor *flow*.



Gambar 2.8 Alur pemodelan *flow*

Pemodelan matematis *flow* ini diturunkan dari model *sump tank* pada Gambar 2.9



Gambar 2.9 Model *sump tank*

Dari model *sump tank* ini didapat persamaan matematis *plant* seperti pada Persamaan (2.27)

$$C dh = (q_i - q_o) dt \quad (2.27)$$

Dari Persamaan 2.27 dapat diturunkan persamaan dari h seperti pada Persamaan 2.28

$$h = q_o \cdot R \quad (2.28)$$

Kemudian dari Persamaan (2.28) ini disubstitusikan ke Persamaan (2.27) seperti pada Persamaan (2.29). Dari Persamaan (2.29) ini kemudian di transformasi *Laplace* sehingga didapatkan Persamaan (2.30)

$$RC \frac{dq_o}{dt} + q_o = q_i \quad (2.29)$$

$$(RC s + 1)Q_o = Q_i \quad (2.30)$$

Dari Persamaan (2.30) kemudian didapatkan persamaan fungsi alih untuk *flow* seperti pada Persamaan (2.31)

$$\frac{Q_o}{Q_i} = \frac{1}{RC s + 1} \quad (2.31)$$

2.5.1 Pemodelan Sensor *Flow*

Elemen terakhir yang perlu dimodelkan sesuai alur pemodelan pada Gambar 2.3 adalah sensor *flow*. Sensor *flow* ini memiliki rentang pembacaan *flow* 0 – 3.5 ltr/menit dengan rentang keluaran 0-10 V. Satuan dari pembacaan *flow* ini dirubah kedalam satuan cm^3/s . Pemodelan untuk sensor *flow* ini seperti pada Persamaan (2.32)

$$K_3 = \frac{\text{Rentang tegangan keluaran sensor (V)}}{\text{Rentang flow air (cm}^3/s)} \quad (2.32)$$

2.6 Kontroler PID [5]

Kontroler PID merupakan salah satu jenis kontroler konvensional yang terbukti handal dan banyak digunakan di industri. Sebagai contoh di Jepang, sekitar 84% dari proses industri menggunakan kontroler PID [6]. Kesederhanaan struktur kontroler PID juga menjadi salah satu alasan mengapa kontroler ini sangat mudah diimplementasikan di mana saja. Kontroler PID memiliki tiga komponen penyusun utama yaitu konstanta proporsional yang dilambangkan dengan K_p , konstanta waktu integral dengan τ_i , dan konstanta waktu differensial dengan τ_d seperti pada Gambar 2.10.



Gambar 2.10 Diagram blok kontroler PID

Dalam persamaan matematis, keluaran dari kontroler PID dapat dirumuskan pada Persamaan (2.33).

$$u(t) = K_p \left[e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{d(e(t))}{dt} \right] \quad (2.33)$$

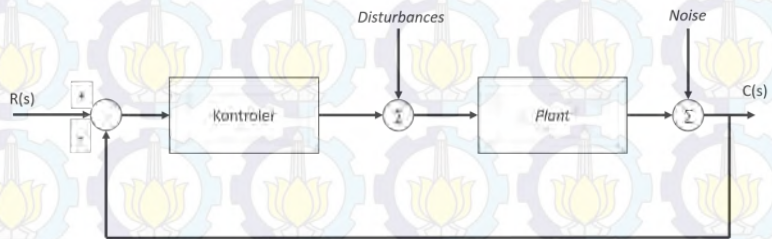
Ketiga konstanta tersebut yang kemudian disebut dengan parameter kontroler PID. Kelebihan dari kontroler PID salah satunya adalah menghilangkan kesalahan keadaan tunak yang belum bisa diatasi apabila hanya menggunakan kontroler proporsional.

Performa kontroler PID sangat ditentukan oleh nilai parameter PID. Untuk memenuhi spesifikasi sistem pengaturan yang diharapkan, parameter PID harus dihitung terlebih dahulu. Ada berbagai macam

metode perhitungan nilai parameter PID, yaitu metode analitis, Zieger Nichols, Cohen-Coon, dan lain sebagainya.

2.7 Disturbances [6]

Dalam sistem pengaturan selalu terdapat gangguan yang dapat mengakibatkan respon dari sistem berubah dari nilai yang diinginkan. Gangguan pada sistem pengaturan in dibedakan menjadi tiga yaitu, perubahan *setpoint* (y_{sp}), gangguan pembebanan (l), dan noise pengukuran (n). Diagram blok dari sistem pengaturan dengan adanya *disturbances* dan *measurement noise* seperti pada Gambar 2.11.



Gambar 2.11 Diagram blok sistem pengaturan dengan *disturbances*

2.7.1 Load Disturbances

Gangguan berupa beban biasanya didominasi gangguan berfrekuensi rendah. Seperti contoh adalah pengaturan kecepatan pada mobil yang bebannya berupa gaya gravitasi yang disebabkan oleh perubahan kemiringan jalan. Gangguan ini berubah secara perlahan dikarenakan perubahan kemiringan jalan berubah secara lambat ketika kita mengendarai mobil. Sinyal *step* dan *ramp* banyak digunakan pada simulasi atau *prototype* sebagai gangguan beban.

2.7.2 Measurement Noise

Berdasarkan pengertiannya, *noise* merupakan komponen yang tidak diinginkan pada sinyal atau variabel. *Measurement noise* dapat merusak informasi mengenai variabel proses yang dikirim oleh sensor yang terdapat pada *plant*. Gangguan berupa *measurement noise* biasanya didominasi gangguan berfrekuensi tinggi. Rata-rata nilai dari daya *spectral noise* ini biasanya nol, jika tidak maka sensor dapat mengirimkan sinyal informasi yang salah atau tidak akurat sehingga menyulitkan

pengaturan pada proses atau *plant*. Salah satu contoh *noise* adalah *Additive White Gaussian Noise* (AWGN)

Noise AWGN merupakan *noise* yang pasti terjadi dalam jaringan nirkabel manapun, memiliki sifat-sifat *additive*, *white*, dan *Gaussian*. Sifat *additive* artinya *noise* ini dijumlahkan dengan sinyal, sifat *white* artinya *noise* tidak bergantung pada frekuensi operasi sistem dan memiliki rapat daya yang konstan, dan sifat *Gaussian* artinya besarnya tegangan *noise* memiliki rapat peluang terdistribusi *Gaussian*.

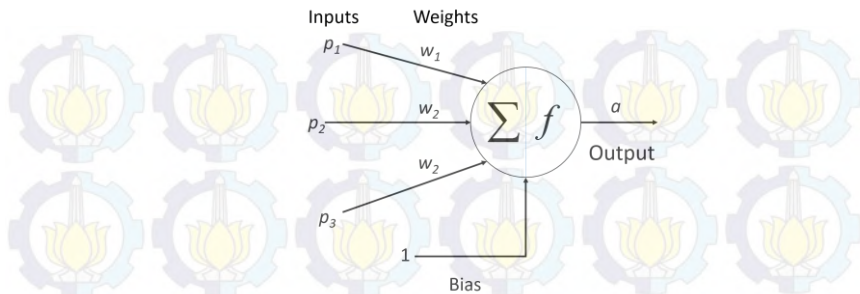
2.8 Jaringan Syaraf Tiruan (*Artificial Neural Network*) [7]

Artificial Neural Network dimulai ketika tahun 1940-an para ilmuwan menemukan bahwa psikologi otak manusia sama dengan mode pemrosesan yang dilakukan oleh komputer. Tahun 1943 McCulloch dan Pitts memperkenalkan model *Neural Network* untuk pertama kalinya. Pada tahun 1954 Farley dan Clark yang pertama kali berhasil menjalankan simulasi komputer *Neural Network*, kemudian pada tahun 1958 Frank Rosenblatt mengembangkan konsep dasar tentang *perceptrons* untuk klasifikasi pola.

Artificial Neural Network bekerja menurut prinsip kerja jaringan syaraf manusia. Sebuah neuron mendapatkan masukan dari dendrit dan memberikan keluaran melalui akson. Informasi masuk kedalam syaraf melalui dendrit. Dendrit ini bertugas mengumpulkan informasi dan meneruskannya ke badan sel. Pada badan sel, informasi yang telah dikumpulkan diolah dengan proses tertentu dan menghasilkan sinyal keluaran. Keluaran dari badan sel diteruskan ke sel syaraf berikutnya melalui akson. Akson ini akan terhubung dengan dendrit sel syaraf berikutnya. Sambungan akson sel syaraf satu dengan dendrit sel syaraf lainnya disebut sinapsis. Kekuatan sambungan pada sinapsis menentukan seberapa besar informasi yang diteruskan oleh sel syaraf sebelumnya dapat diterima oleh sel syaraf berikutnya, yang kemudian disebut dengan bobot.

2.8.1 Konsep Dasar Pemodelan *Neural Network*

Tiruan neuron dalam struktur *Neural Network* adalah sebagai elemen proses yang dapat berfungsi seperti halnya sebuah neuron. Model dari neuron pada *Neural Network* ini seperti pada Gambar 2.12, dimana p_i merupakan masukan pada neuron dan w_i merupakan bobot atau *weight*. Sinyal masukan p_i ini akan dikalikan dengan nilai w_i . Nilai keluaran dari neuron ini dapat dirumuskan seperti pada Persamaan (2.34)



Gambar 2.12 Model neuron

$$a = f(z) = f\left(\left(\sum_{i=1}^n p_i w_i\right) - \theta\right) \quad (2.34)$$

Dimana θ merupakan nilai bias, dan f merupakan fungsi aktivasi pada neuron. Tingkat aktivasi dari neuron dapat berharga diskrit yaitu 0 dan 1, atau kontinu yaitu antara 0 dan 1. Hal tersebut bergantung pada fungsi aktivasi yang digunakan. Ada berbagai macam fungsi aktivasi yang digunakan yaitu:

- 1) *Threshold function*
- 2) *Hard limit function*
- 3) *Sigmoid function*
- 4) *Linear Function*

Dari macam-macam fungsi aktivasi pada neuron, yang paling sering digunakan adalah *sigmoid function* karena dianggap lebih mendekati dari kinerja sinyal pada otak manusia.

2.8.2 Learning pada Neural Network

Learning pada ANN dilakukan dengan merubah bobot agar mencapai target yang diinginkan. Ada tiga metode dalam *learning*, yaitu.

- a. *Supervised learning*
Supervised learning adalah metode pembelajaran yang diarahkan untuk mempelajari contoh-contoh yang telah disediakan.
- b. *Unsupervised learning*

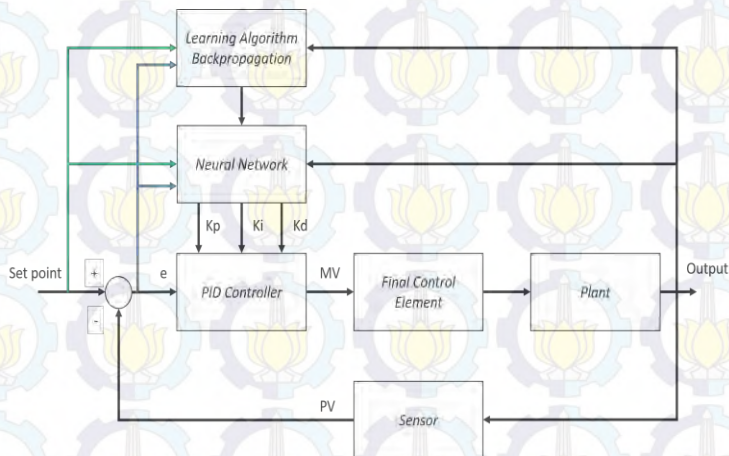
Unsupervised learning mempelajari pola informasi dari masukan namun tidak ada *feedback* yang secara tegas menjelaskan target yang akan dicapai.

c. *Reinforced learning*

Reinforced learning mempelajari pola masukan dan keluaran berdasarkan konsep reward and punishment. Jika keluaran bergerak menuju target yang diinginkan, maka diberikan *reward*. Namun jika keluaran bergerak menjauhi target yang diinginkan, maka diberikan *punishment*.

2.9 PID – Neural Network

Kontroler PID-*Neural Network* terdiri atas kontroler PID sebagai kontroler utama dan *Neural Network* sebagai algoritma cerdas yang dapat melakukan *tuning* parameter kontroler PID dengan diagram blok seperti pada Gambar 2.14



Gambar 2.13 Diagram blok sistem pengaturan dengan kontroler PID *Neural Network*

Algoritma dari kontroler PID berbasis *backpropagation* ini dapat dituliskan sebagai berikut:

- 1) Memilih struktur dari *backpropagation Neural Network*, jumlah *layer*, jumlah neuron pada setiap *layer*, memberi inisialisasi *weight*

pada jaringan setiap *layer*, memilih *learning rate* dan koefisien inersia

- 2) Mendapatkan nilai $r(k)$ *reference* dari dan $e(k)$ dari sampling, dan kemudian menghitung *error* $e(k)=r(k)-y(k)$, dimana $i=k, k-1, \dots, k-p$
- 3) Normalisasi dari setiap nilai proses $r(i)$, $y(i)$, $u(i-1)$ dan $e(i)$ yang kemudian digunakan sebagai masukan pada *Neural Network*
- 4) Menghitung setiap nilai pada masukan *value* dan keluaran *value* setiap *layer* pada *Neural Network*, dan *output layer* dari *Neural Network* sebagai nilai parameter K_p , K_i , dan K_d
- 5) Melakukan training secara on-line, dengan merubah bobot pada setiap neuron dengan menggunakan algoritma pembelajaran *Backpropagation*
- 6) Pada iterasi berikutnya atau $k=k+1$ maka kembali ke langkah nomor 2

Pada domain waktu kontinyu Persamaan kontroler PID seperti pada Persamaan (2.35)

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{d(t)}{dt} \right] \quad (2.35)$$

Karena kontroler ini akan diproses di komputer maka diubah kedalam bentuk diskrit seperti pada Persamaan (2.36)

$$u(n) = u(n-1) + K_p(e(n) - e(n-1)) + K_i e(n) + K_d(e(n) - 2e(n-1) + e(n-2)) \quad (2.36)$$

Dengan K_p , K_i , dan K_d merupakan *proportional*, *integral*, dan *derivative gain*. $U(n)$ melambangkan masukan *plant* pada nT dan $e(n)$ merupakan *error* antara nilai yang diinginkan atau *set point* dengan nilai aktual dari sistem seperti pada Persamaan (2.37)

$$e(n) = r(n) - y(n) \quad (2.37)$$

Setiap *layer* terdiri atas N_1 , N_2 , dan N_3 neuron, dimana untuk neuron N_1 , N_2 dapat dipilih secara *trial* dan *error* tergantung pada kompleksitas dari *plant* dan N_3 bernilai sama dengan jumlah parameter PID. Metode

Backpropagation ini memerlukan nilai E atau indeks performansi yang perlu diminimalkan. Persamaan indeks performansi yang digunakan seperti pada Persamaan (2.38).

$$E = \frac{1}{2} (r(n) - y(n))^2 \quad (2.38)$$

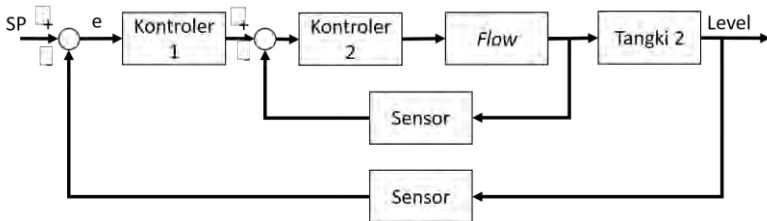
BAB 3

PERANCANGAN SISTEM

Pada Bab ini dibahas mengenai perancangan sistem pengaturan *level* fluida pada simulator *plant coupled tank*. Dimulai dari pemodelan *plant coupled tank*, perancangan kontroler PID dengan *root locus* dan kontroler PID- *Neural Network*, dan perancangan HMI.

3.1 Gambaran Umum Sistem

Sistem yang akan dibangun pada penelitian ini merupakan sistem pengaturan *level cascade* dengan *flow* yang akan diterapkan pada simulator *plant coupled tank*. Diagram blok dari sistem ini seperti pada Gambar 3.1. Dalam pengaturan *cascade* terdapat *inner loop* dan *outer loop*. Untuk sistem pengaturan *cascade level* ini yang menjadi objektif kontrol adalah pengendalian *level liquid* pada tangki 2. Maka proses *level liquid* berada pada *outer loop*. Pada *outer loop* ini kontroler yang digunakan adalah PID-*Neural Network*. Sistem pengaturan *flow* diletakkan pada *inner loop* dikarenakan juga proses *flow* memiliki respon yang lebih cepat dan pada aplikasinya *level liquid* pada tangki dipengaruhi oleh *flow* yang masuk pada tangki. Pada *inner loop* kontroler yang akan digunakan adalah kontroler PID.



Gambar 3.1. Diagram blok sistem pengaturan *level plant coupled tank*

Simulator *plant coupled tank* ini akan dibangun menggunakan perangkat lunak LabVIEW dengan menggunakan tambahan modul Control & Simulation dan DCS. Simulator ini akan dibangun berdasarkan permodelan matematis dari *plant*. Pada perangkat lunak LabVIEW ini tidak hanya dibangun simulator dari *plant* ini saja, tetapi algoritma kontrol yang akan digunakan juga dibangun di LabVIEW.

3.2 Pemodelan Sistem Pengaturan *Level Plant Coupled Tank*

3.2.1 Pemodelan Pompa

Pompa yang digunakan pada *plant* ini adalah pompa air DC. Rentang debit air yang ada dalam satuan liter/menit diubah dulu kedalam satuan sehingga didapatkan rentang debit air keluar dari pompa sebesar 0-58.33. Dengan memasukkan nilai parameter ke Persamaan (2.1) pemodelan pompa ini didapatkan nilai K_1 seperti pada Persamaan (3.1)

$$K_1 = \frac{58.33-0}{10-0} = 5.83 \text{ cm}^3 / sV \quad (3.1)$$

3.2.2 Pemodelan Pipa

Diameter pipa yang digunakan pada sistem ini sebesar 1 cm dengan panjang 90 cm dan debit air yang masuk sebesar $16.67 \text{ cm}^3 / s$. Dari data ini kemudian dimasukkan pada Persamaan (2.2) didapat waktu tunda seperti pada Persamaan (3.2)

$$\theta = \frac{70.65}{16.67} = 4.23s \quad (3.2)$$

3.2.3 Pemodelan Tangki

Dalam pemodelan tangki *coupled tank* ini dibutuhkan nilai parameter dari R_1 , R_2 , R_3 , C_1 dan C_2 . Nilai dari parameter ini seperti tertera pada Tabel 3.1

Tabel 3.1 Nilai Parameter R dan C pemodelan tangki

Simbol	Nilai	Satuan
R_1	0.5	s / cm^2
R_2, R_3	0.1	s / cm^2
C_1, C_2	289	cm^2

Setelah mendapatkan nilai parameter R dan C maka nilai tersebut dimasukkan pada Persamaan (2.21) dan (2.25) maka didapatkan fungsi alih dari sistem seperti pada Persamaan (3.3) dan (3.4)

$$\frac{H_1(s)}{Q_i(s)} = G_1(s) = \frac{144.5s + 6}{41761s^2 + 3468s + 20.5} \quad (3.3)$$

$$\frac{H_2(s)}{Q_i(s)} = G_2(s) = \frac{1}{41761s^2 + 3468s + 20.5} \quad (3.4)$$

3.2.4 Pemodelan Sensor Level

Sensor yang digunakan pada *plant* ini adalah sensor *level* dengan tipe *floating*. Dari pemodelan sensor ini akan didapat nilai *gain* atau penguatan K_2 . K_2 dirumuskan berupa perbandingan rentang kerja dari pompa terhadap masukan berupa tegangan seperti pada Persamaan (2.26)

Dengan memasukkan rentang tegangan keluaran antara 0 – 10 V dan rentang ketinggian air dalam tangki antara 0 – 18 cm didapatkan pemodelan sensor level seperti pada Persamaan (3.5).

$$K_2 = \frac{10-0}{18-0} = 0.55 \text{ V/cm} \quad (3.5)$$

3.2.5 Fungsi Alih Keseluruhan

Berdasarkan pemodelan dari setiap komponen penyusun sistem ini seperti pada Gambar 1.2 didapatkan fungsi alih keseluruhan dari sistem seperti pada Persamaan (3.7) untuk $G_1(s)$ dan (3.8) untuk $G_2(s)$

$$G_{total}(s) = K_1.G(s).K_2.e^{-ts} \quad (3.6)$$

$$G_{1total}(s) = \frac{463.34s + 19.24}{41761s^2 + 3468s + 20.5} e^{-4.23s} \quad (3.7)$$

$$G_{2total}(s) = \frac{3.207}{41761s^2 + 3468s + 20.5} e^{-4.23s} \quad (3.8)$$

3.2.6 Pemodelan Sistem Pengaturan Flow Plant Coupled Tank

Pada pemodelan matematis *flow* ini nilai R yang digunakan sebesar 0.5 dan C yang digunakan adalah luas penampang pipa sebesar 0.78,

kemudian nilai tersebut dimasukkan pada Persamaan (2.31) sehingga didapatkan seperti pada Persamaan (3.9)

$$\frac{Q_o}{Q_i} = \frac{1}{0.392s + 1} \quad (3.9)$$

3.2.6.1 Pemodelan Sensor Flow

Sensor *flow* ini memiliki rentang pembacaan *flow* 0 – 3.5 ltr/menit dengan rentang keluaran 0-10 V. Satuan dari pembacaan *flow* ini dirubah kedalam satuan cm^3/s . Nilai tersebut kemudian dimasukkan pada persamaan (2.32) sehingga didapatkan penguatan K_3 seperti pada Persamaan (3.10)

$$K_3 = \frac{0 - 10(V)}{53.88 - 0(cm^3/s)} = 0.171 \quad (3.10)$$

3.2.6.2 Fungsi Alih Keseluruhan Proses Flow

Dari semua elemen kemudian didapatkan fungsi alih keseluruhan dari sistem seperti pada Persamaan (3.11)

$$G_{total}(s) = K_1.G(s).K_3.e^{-t_k} \quad (3.11)$$

Untuk nilai K_1 digunakan nilai penguatan dari pompa seperti pada persamaan (3.2) sehingga didapat fungsi alih keseluruhan dari proses *flow* seperti pada Persamaan (3.12)

$$\frac{Q_o}{Q_i} = \frac{0.92}{0.392s + 1} e^{-4.23s} \quad (3.12)$$

3.3 Perancangan Kontroler

3.3.1 Perancangan Kontroler PID dengan Root Locus

Spesifikasi desain dari kontroler PID ini adalah *overshoot* 0% dan kesalahan keadaan tunak sama dengan 0. Pertama fungsi alih dari *delay* pada sistem dirubah seperti Persamaan (3.13) dengan menggunakan orde pertama dari deret Taylor

$$e^{-\theta s} = 1 - \theta s + \frac{\theta^2 s^2}{2!} - \frac{\theta^3 s^3}{3!} + \frac{\theta^4 s^4}{4!} \dots \quad (3.13)$$

Sehingga fungsi alih dari sistem G_1 menjadi seperti pada Persamaan (3.14)

$$G_{1total}(s) = \frac{K(1-\theta s)}{(s+\alpha)(s+\beta)} \quad (3.14)$$

Dimana $(s+\alpha)(s+\beta)$ adalah persamaan karakteristik dari fungsi alih.

Kontroler PID yang digunakan seperti pada Persamaan (3.15)

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left[\frac{s + \frac{1}{\tau_i} + \tau_d s^2}{s} \right] \quad (3.15)$$

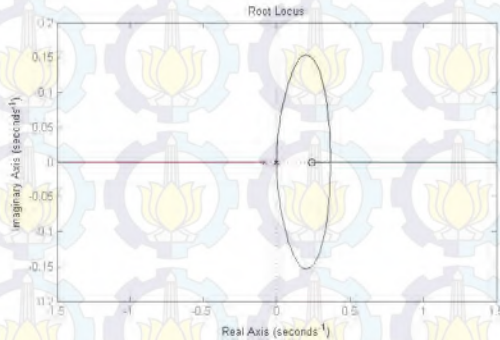
Pada tahap ini dilakukan perancangan kontroler PID untuk sistem pengaturan *level* pada tangki dua. Persamaan keseluruhan kontroler dari *plant* ini seperti pada Persamaan (3.16)

$$G_c(s)G_{1total} = K_p \left[\frac{s + \frac{1}{\tau_i} + \tau_d s^2}{s} \right] \left[\frac{0.157(1-4.23s)}{2037s^2 + 173.4s + 1} \right] \quad (3.16)$$

$$G_c(s)G_{1total} = K_p \left[\frac{s + \frac{1}{\tau_i} + \tau_d s^2}{s} \right] \left[\frac{-0.6641s + 0.157}{(s + 0.07655)(s + 0.006413)} \right] \quad (3.17)$$

Dari Persamaan (3.17) dapat diketahui *pole* dari *plant* ini. Untuk *pole* terletak pada $s_1=0$, $s_2=-0.006413$, $s_3=-0.07655$. Dari Persamaan (3.17) didapat *root locus* dari sistem seperti pada Gambar 3.2. Pada

Gambar 3.2 dapat diamati bahwa *locus* banyak yang berada di sebelah kanan sumbu imajiner, sehingga dapat membuat sistem tidak stabil. Sehingga diperlukan penambahan *zero* untuk dapat menarik *locus* ke sebelah kiri sumbu imajiner. Dengan digunakannya kontroler PID maka akan ditambahkan dua *zero* ke dalam *root locus* dari *plant* ini.



Gambar 3.2 Root locus dari sistem G_2

Kemudian akan ditambahkan *zero* untuk nilai z_1 diantara nilai *pole* $s_1=0$ dan $s_2=-0.00621$. Dipilih letak $z_1=-0.0061$ dan untuk z_2 dipilih pada titik $z_2=-0.2$. Kemudian untuk penentuan *gain* dipilih menggunakan diagram *root locus* seperti pada Gambar 3.3.



Gambar 3.3 Root locus plant setelah penambahan kontroler PID

Dipilih nilai *gain* sebesar 100 karena berdasarkan *root locus*, nilai *gain* tersebut memiliki *overshoot* sebesar 0%. Sehingga parameter kontroler PID yang didapatkan yaitu $\tau_i = 158.98$, $K_p = 100 \times 0.2061 = 20.61$, dan $\tau_d = 4.85$. Persamaan kontroler PI yang digunakan seperti pada Persamaan (3.18).

$$G_c(s) = \frac{U(s)}{E(s)} = 20.61 \left[1 + \frac{1}{158.98s} + 4.85s \right] \quad (3.18)$$

3.3.2 Perancangan Kontroler PID dengan Metode Miluse Viteckova

Metode ini digunakan untuk *plant* dengan jenis *First Order Plus Dead Time (FOPDT)*. Pada metode ini pertama kali diidentifikasi parameter dari fungsi alih *plant*. Persaman umum dari *FOPDT* pada metode ini seperti pada persamaan (3.19).

$$G_p(s) = \frac{k_1}{T_1s + 1} e^{-T_d s} \quad (3.19)$$

Tabel 3.2 Tuning Parameter Kontroler PID dengan metode Miluse Viteckova

Tipe		Kontroler
		Analog T=0, Digital T>0
PI	T^*_I	$T_1 - \frac{T}{2}$
	K^*_p	$\frac{T_1^*}{k_1[(4-e)T + eT_d]}$
PID	T^*_I	$\frac{2(T_d + T)(2T_1 - T) + T_d^2}{4(T_d + T)}$
	K^*_p	$\frac{4T_1^*}{k_1[(14 - e^2)T + e^2T_d]}$
	T^*_D	$\frac{(2T_1 - T)T_d^2}{8(T_d + T)T_1^*}$

Dari fungsi alih *plant* pada persamaan (3.12) didapatkan parameter $k_l = 0.92$, $T_l = 0.392$, dan $T_d = 4.32$. Untuk perhitungan parameter kontroler PID pada metode ini sesuai dengan Tabel 3.2. Spesifikasi dari metode ini adalah kesalahan keadaan tunak sama dengan 0 dan tanpa *overshoot*.

Kemudian dilakukan perhitungan parameter kontroler PID seperti pada persamaan (3.20), (3.21), dan (3.22) dipilih nilai $T=0$ karena kontroler yang digunakan adalah kontroler analog.

$$T_1^* = \frac{2(T_d + T)(2T_l - T) + T_d^2}{4(T_d + T)} = \frac{25.4362}{17.28} = 1.4720 \quad (3.20)$$

$$K_p^* = \frac{4T_1^*}{k_l[(14 - e^2)T + e^2T_d]} = \frac{5.880}{29.3672} = 0.2002 \quad (3.21)$$

$$T_D^* = \frac{(2T_l - T)T_d^2}{8(T_d + T)T_1^*} = \frac{14.6313}{50.8723} = 0.2876 \quad (3.22)$$

Sehingga didapatkan fungsi alih kontroler seperti pada Persamaan (3.23)

$$G_c(s) = \frac{U(s)}{E(s)} = 0.2002 \left[1 + \frac{1}{1.4720s} + 0.2876s \right] \quad (3.23)$$

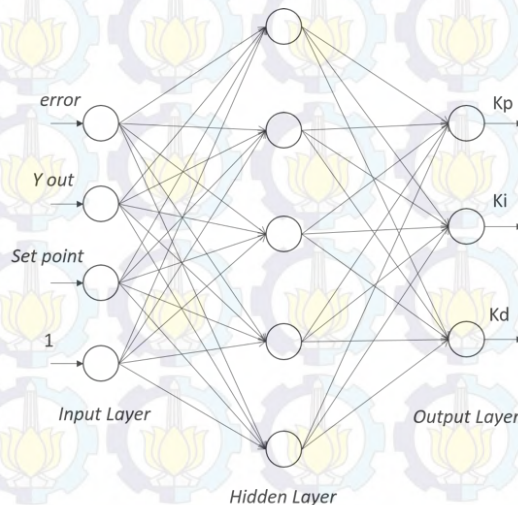
3.3.3 Perancangan Kontroler PID-Neural Network

Kontroler PID-*Neural Network* terdiri atas kontroler PID sebagai kontroler utama dan *Neural Network* sebagai algoritma cerdas yang dapat melakukan *tuning* parameter kontroler PID. Dalam mendesain kontroler PID-*Neural Network* adapun langkahnya sebagai berikut:

1. Menentukan algoritma pembelajaran yang akan digunakan pada *Neural Network*, algoritma pembelajaran yang digunakan pada perancangan kontroler ini adalah *backpropagation*.
2. Menentukan struktur dari *Neural Network*, seperti jumlah *layer* yang digunakan dan jumlah neuron pada setiap *layer*

3. Melakukan inisialisasi nilai bobot yang digunakan pada *Neural Network* pada bagian antara input layer menuju hidden layer dan hidden layer menuju output layer
4. Menentukan nilai *learning rate* (η) dan koefisien inersia (α)

Pada kontroler PID Neural Network yang dirancang terdiri atas tiga layer, yaitu satu *input layer*, satu *hidden layer*, dan satu *output layer*. Jumlah neuron yang digunakan pada *input layer* terdiri atas empat neuron, pada *hidden layer* lima neuron, dan tiga neuron pada *output layer* yang merepresentasikan dari tiga parameter kontroler PID yaitu K_p , K_i , dan K_d . Struktur dari *Neural Network* yang digunakan seperti pada Gambar 3.4.



Gambar 3.4 Struktur *Neural Network* untuk autotuning PID

Pada *Neural Network* ini terdapat mekanisme algoritma *feed forward* dan algoritma *backpropagation*. Pada algoritma *feed forward* dilakukan perhitungan maju dari *input layer* menuju *output layer* untuk mendapatkan nilai keluaran dari *Neural Network*. Nilai keluaran ini kemudian dibandingkan dengan nilai referensi yang digunakan.

Pada algoritma *feed forward* ini pertama dibuat persamaan untuk masukan pada *input layer* seperti pada Persamaan (3.24)

$$O_j^1 = x(j) \quad (j = 1, 2, \dots, M) \quad (3.24)$$

M merupakan jumlah variabel masukan pada *input layer*, pada PID-*Neural Network* ini digunakan empat variabel masukan yaitu *error* yaitu selisih antara nilai referensi dan keluaran dari *plant*, *Y out* yaitu keluaran dari *plant*, *setpoint* atau nilai referensi yang diberikan pada *plant* dan nilai konstanta 1. Masukan pada *hidden layer* dapat dituliskan seperti pada Persamaan (3.24) dan keluaran dari *hidden layer* seperti pada Persamaan (3.25)

$$net_1^{(2)} = \sum_{j=0}^M w_{ij}^{(2)} O_j^{(1)} \quad (3.25)$$

$$O_i^2(k) = f(net_1^{(2)}(k)) \quad (i = 1, 2, \dots, Q) \quad (3.26)$$

Q merupakan jumlah neuron yang terdapat pada *hidden layer*, dan f merupakan fungsi aktivasi pada *hidden layer*, w_{ji} merupakan nilai bobot pada jaringan antara *input layer* menuju *hidden layer*. Fungsi aktivasi yang digunakan pada *hidden layer* ini adalah *tanh sigmoid* seperti pada Persamaan (3.26)

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (3.27)$$

Untuk masukan dan keluaran pada *output layer* dapat dituliskan dalam persamaan matematis seperti pada Persamaan (3.27) dan (3.28)

$$net_1^{(3)}(k) = \sum_{i=0}^Q w_{ij}^{(3)} O_i^{(2)}(k) \quad (3.28)$$

$$O_l^{(3)}(k) = g(net_1^{(3)}(k)) \quad (l = 1, 2, 3) \quad (3.29)$$

Dimana g merupakan fungsi aktivasi yang digunakan pada *output layer*. Karena nilai dari $O_l^{(3)}(k)$ tidak boleh bernilai negatif maka digunakan fungsi aktivasi *non-negative sigmoid* seperti pada Persamaan (3.30)

$$g(x) = \frac{e^x}{e^x + e^{-x}} \quad (3.30)$$

Keluaran dari *output layer* ini merupakan nilai parameter kontroler PID yaitu K_p , K_i , dan K_d seperti pada Persamaan (3.31)

$$\begin{aligned} O_1^{(3)}(k) &= K_p \\ O_2^{(3)}(k) &= K_i \\ O_3^{(3)}(k) &= K_d \end{aligned} \quad (3.31)$$

Kemudian dilakukan mekanisme *Backpropagation* untuk perubahan nilai bobot pada *Neural Network*. Pada algoritma ini fungsi *error* atau indeks performa pada keluaran sistem didefinisikan seperti pada Persamaan (3.32)

$$E(k) = \frac{1}{2} (r(k) - y(k))^2 = \frac{1}{2} e^2(k) \quad (3.32)$$

Perubahan bobot pada *hidden layer* menuju *output layer* seperti pada Persamaan (3.33)

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} + \alpha w_{li}^{(3)}(k-1) \quad (3.33)$$

Dimana η merupakan *learning rate*, dan α merupakan *momentum factor*. *Learning rate* ini merupakan parameter laju pembelajaran pada algoritma *Neural Network* yang mempengaruhi kecepatan algoritma *Neural Network* mendapatkan nilai solusi minimum.

Algoritma pembelajaran untuk merubah bobot pada *output layer* dapat dituliskan seperti pada Persamaan (3.34), (3.35), dan (3.36)

$$w_{li}^{(3)}(k+1) = w_{li}^{(3)}(k) + \Delta w_{li}^{(3)}(k) \quad (3.34)$$

$$\Delta w_{li}^{(3)}(k) = \alpha \Delta w_{li}^{(3)}(k-1) + \eta \delta_l^{(3)} O_i^{(2)}(k) \quad (3.35)$$

$$\delta_l^{(3)} = e(k) \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial O_i^{(3)}(k)} \cdot g'(in_l^{(3)}(k)) \quad (3.36)$$

Dimana turunan pertama dari $g(x)$ seperti pada Persamaan (3.37)

$$g'(x) = g(x)(1 - g(x)) \quad (3.37)$$

Untuk algoritma pembelajaran perubahan bobot pada *hidden layer* seperti pada Persamaan (3.38), (3.39), dan (3.40)

$$w_{ij}^{(3)}(k+1) = w_{ij}^{(2)}(k) + \Delta w_{ij}^{(2)}(k) \quad (3.38)$$

$$\Delta w_{ij}^{(2)}(k) = \alpha \Delta w_{ij}^{(2)}(k-1) + \eta \delta_i^{(2)} O_j^{(1)}(k) \quad (3.39)$$

$$\delta_i^{(2)} = f'(in_i^{(2)}(k)) \cdot \sum_{l=1}^3 \delta_l^{(3)} w_{li}^{(3)}(k) \quad (3.40)$$

Dalam *Neural Network* ini perlu dilakukan inisialisasi bobot. Inisialisasi bobot antara masukan dan hidden layer mengikuti persyaratan pemetaan $(r,y) \rightarrow e$ seperti pada Persamaan (3.41) dan (3.42), sedangkan bobot antara *hidden layer* dan *output layer* sebaiknya mengambil nilai positif terkecil [7].

$$r \rightarrow \text{hidden layer} \quad w_{s1j}(0) = +1 \quad (3.41)$$

$$y \rightarrow \text{hidden layer} \quad w_{s2j}(0) = -1 \quad (3.42)$$

Inisialisasi *weight* diberikan antara *input layer* dengan *hidden layer* seperti yang ditunjukkan pada Tabel 3.3.

Tabel 3.3 Inisiasi *weight input layer* menuju *hidden layer*

	w_{1i}	w_{2i}	w_{3i}	w_{4i}
w_{j1}	-0.6394	-0.2696	-0.3756	-0.7023
w_{j2}	-0.8603	-0.2013	-0.5024	-0.2596
w_{j3}	-1.0749	0.5543	-1.6820	-0.5437
w_{j4}	-0.3625	-0.0724	-0.6463	-0.2859
w_{j5}	0.1425	0.0279	-0.5406	-0.7660

Inisialisasi *weight* antara *hidden layer* dan *output layer* seperti pada Tabel 3.4. Dengan keterangan w_{ji} , j merupakan neuron dimana *weight* itu berasal dan i merupakan jumlah neuron yang dituju *weight* tersebut.

Tabel 3.4 Inisiasi *weight hidden layer* menuju *output layer*

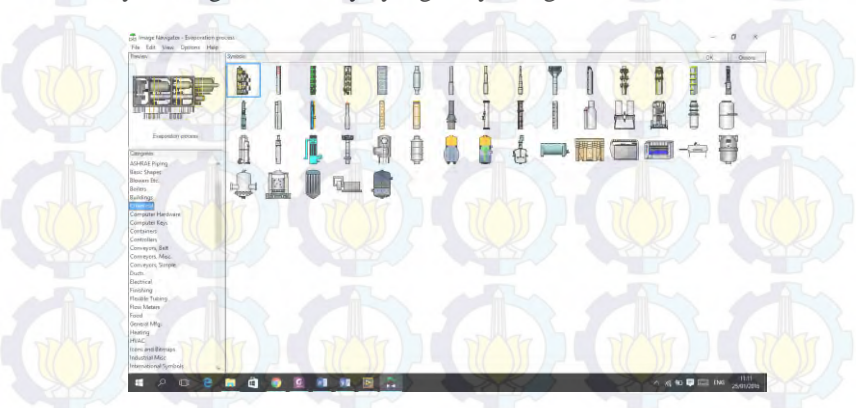
	w_{1i}	w_{2i}	w_{3i}	w_{4i}	w_{5i}
w_{j1}	0.7576	0.2616	0.5820	-0.1416	-0.1325
w_{j2}	-0.1146	0.2949	0.8352	0.2205	0.4508
w_{j3}	0.7201	0.4566	-0.7672	0.4962	0.3632

3.4 Perancangan *Human Machine Interface* (HMI)

Human Machine Interface (HMI) merupakan antarmuka antara operator dengan *plant* yang dikendalikan. HMI ini berfungsi untuk memudahkan operator untuk berinteraksi dengan proses. Pada umumnya tampilan HMI dibuat dengan *screen* terpisah-pisah dengan fungsi masing-masing, mulai dari fungsi *monitoring*, *controlling*, *reporting* dan *alarm*

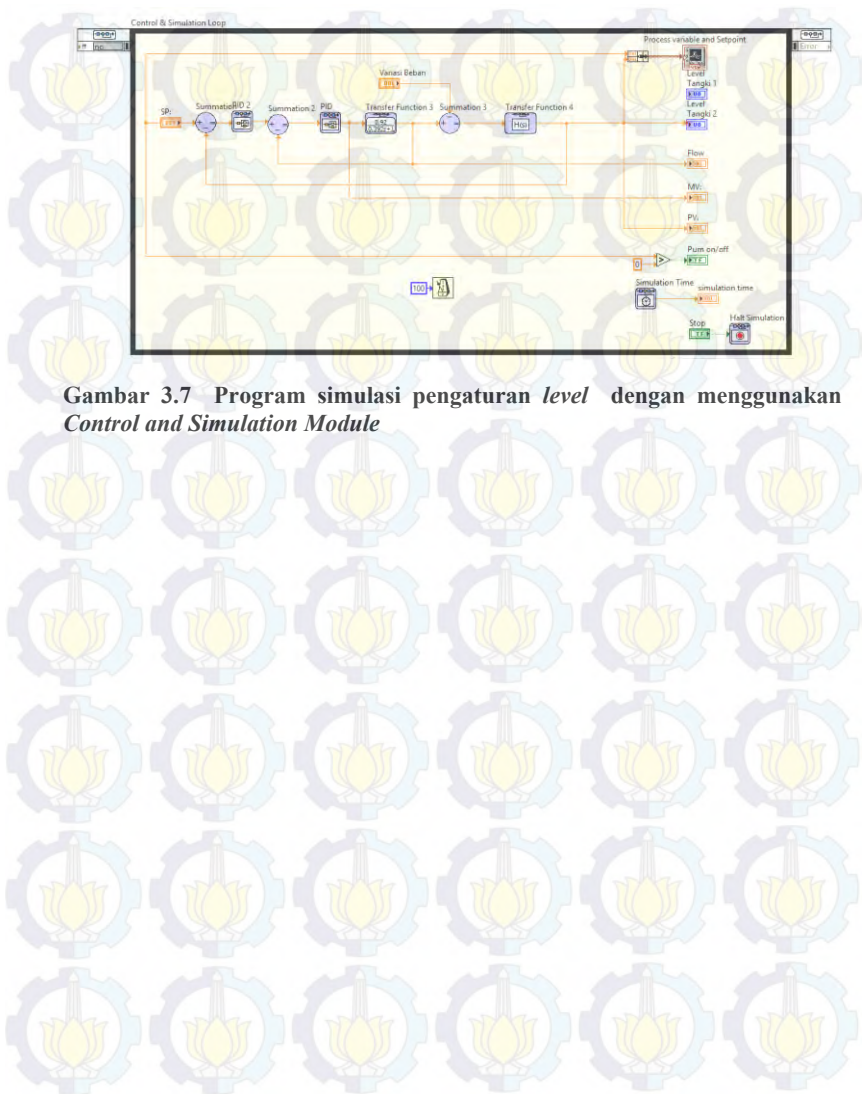
Pada perancangan HMI coupled tank ini digunakan perangkat lunak LabVIEW. Keuntungan dari perangkat lunak LabVIEW ini pembuatan HMI dan program yang akan disambungkan ke HMI sudah dalam satu program. Perangkat lunak LabVIEW ini terdiri atas dua bagian yaitu *front panel* dan *block diagram*. *Front panel* ini berfungsi sebagai UI (User Interface) / HMI, sedangkan *block diagram* merupakan tempat dimana program LabVIEW dibuat.

Perangkat lunak LabVIEW standar sudah menyediakan gambar seperti *knob*, *led*, *grap chart*, *slider* untuk pembuatan HMI pada front panel, tetapi untuk gambar tangki, sensor, pipa, pompa, dan perangkat aktuator dan instrumen lainnya belum tersedia, sehingga diperlukan menambah *add on DCS Module* pada perangkat lunak LabVIEW ini. HMI yang akan dibuat ini merepresentasikan dari *plant coupled tank*. Penggambaran dari *plant coupled tanks* ini terdiri atas dua tangki vertikal dan satu tangki penyimpanan, untuk sensor dan aktuator yang ditampilkan adalah satu sensor *flow* dan tiga *pneumatic control valve*. Pemilihan gambar dari tangki, sensor dan aktuator melalui menu *image navigator* pada DSC Module. Menu ini terdapat pada *Tools* → *DSC Module* → *Image Navigator*. Pada menu *image navigator* seperti pada Gambar 3.5 terdapat berbagai macam gambar dari mulai boiler, sensor, tangki, pompa, conveyor dan gambar lainnya yang banyak digunakan di HMI.



Gambar 3.5 Image Navigator pada DSC Module

Indikator bar dan numerik ini disambungkan pada *output* dari model matematis *plant*. Pada sensor *flow* diberi indikator numerik untuk mengetahui *flow* fluida yang lewat, sedangkan pada pompa diberi indikator led. Indikator led ini akan berwarna merah ketika pompa mati dan akan menyala berwarna hijau ketika pompa hidup.



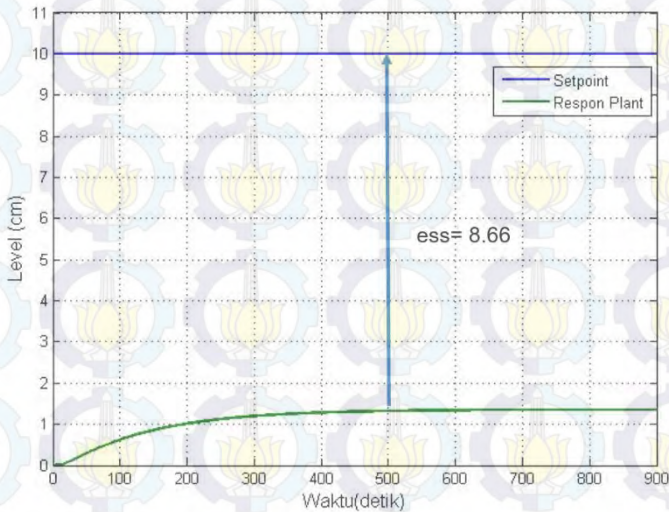
Gambar 3.7 Program simulasi pengaturan *level* dengan menggunakan *Control and Simulation Module*

BAB 4 PENGUJIAN DAN ANALISA

4.1 Simulasi Sistem Tanpa Kontroler

4.1.1 Respon Closed Loop Proses Level

Pada tahap ini dilakukan simulasi sistem *plant* coupled tank pada tangki 2. Dari persamaan fungsi alih keseluruhan G_2 pada Persamaan (3.37) dilakukan pengujian *closed loop* tanpa diberi kontroler terlebih dahulu. Sinyal uji yang diberikan berupa sinyal step dengan nilai *set point* sepuluh. Hasil simulasi didapatkan respon sistem seperti pada Gambar 4.1



Gambar 4.1. Respon closed loop sistem

Dari respon sistem pada Gambar 4.1 dapat diamati bahwa sistem G_2 memiliki *rise time* sebesar 119 detik. Dari pengujian ini didapat fungsi alih *closed loop* G_2 seperti pada persamaan (4.1)

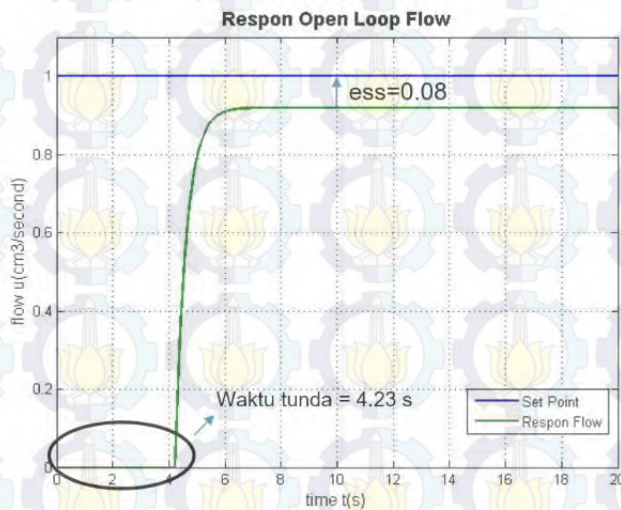
$$G_{2total}(s) = \frac{-95.6s^2 + 18.66s + 0.93}{1941s^2 + 192.1 + 1.93} \quad (4.1)$$

Dari persamaan (4.1) Untuk sistem G_2 didapat nilai $\omega_n = 0.0315$ dan nilai $\zeta = 1.5682$. Untuk sistem G_2 ini memiliki respon *overdamped* dikarenakan memiliki nilai $\zeta > 1$. Respon pada gambar 4.1 untuk G_2 ini sesuai dengan perhitungan nilai ζ bahwa memiliki respon overdamped.

Respon yang dihasilkan untuk sistem G_2 masih memiliki kesalahan keadaan tunak yang cukup besar, yaitu sebesar 8.66. Maka diperlukan kontroler yang dapat menghilangkan kesalahan keadaan tunak.

4.1.2 Respon *Open Loop* Proses *Flow*

Pada tahap ini dilakukan simulasi *open loop* pada proses *flow* untuk mengetahui respon dari *plant*. Simulasi *open loop* pada *plant flow* ini dengan diberi masukan berupa unit step. Dari hasil simulasi didapat respon dari *plant* seperti pada Gambar 4.2.



Gambar 4.2 Respon *open loop* proses *flow*

Dari hasil respon masih terdapat kesalahan keadaan tunak sebesar 0.08 dan juga waktu tunda sebesar 4.23 detik. Waktu tunda ini bisa disebabkan mengalirnya air dalam pipa menuju ke tangki. Sehingga apabila dalam *real system* semakin panjang pipa yang dilalui fluida maka waktu tunda akan semakin besar juga. Berdasarkan hasil respon maka

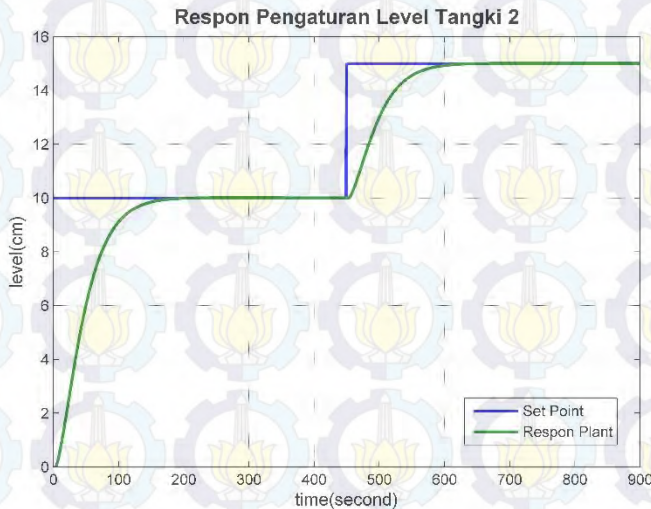
diperlukan kontroler yang mampu menghilangkan kesalahan keadaan tunak terlebih dahulu.

4.2 Simulasi Sistem dengan Kontroler PID

Pada tahap ini dilakukan pengujian pada *plant coupled tank* dengan fungsi alih G_2 dengan konfigurasi *single loop* dan *cascade*. Sinyal uji yang diberikan berupa sinyal step dan kontroler yang digunakan adalah kontroler PID hasil rancangan dengan menggunakan *root locus* yang tertera pada Bab 3. Pengujian pada sistem ini meliputi:

1. Pengujian dengan variasi *set point*
2. Pengujian dengan variasi parameter kontroler (K_p , τ_i , dan τ_d)
3. Pengujian dengan diberi beban
4. Pengujian dengan *noise*

4.2.1 Simulasi Pengaturan Level Single loop Variasi Set point



Gambar 4.3 Respon *plant* dengan perubahan *setpoint*

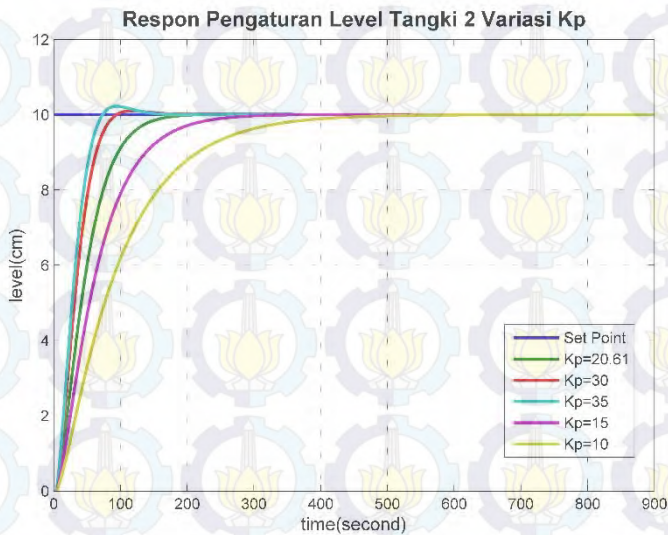
Pada simulasi pengujian dengan perubahan *set point* ini digunakan sinyal step. *Set point* awal diberikan nilai sepuluh hingga detik ke 450 kemudian *set point* dinaikkan hingga bernilai 15 sampai detik ke 900.

Pengujian terhadap *set point* ini bertujuan untuk membuktikan kinerja kontroler terhadap perubahan nilai referensi yang diberikan.

Simulasi pengujian ini langsung dilakukan untuk pengaturan *level* pada tangki 2 seperti pada Gambar 4.3. Pada Gambar 4.3 waktu yang dibutuhkan oleh respon pengaturan *level* pada tangki 2 untuk mencapai nilai 63.2% dari keadaan tunak atau 6.32 yaitu sebesar 49.62 detik, Dari respon pada Gambar 4.3 dapat diamati bahwa kontroler PID mampu mengikuti perubahan setpoint yang diberikan dan kontroler PID mampu menghilangkan kesalahan keadaan tunak pada pengaturan *level* tangki 2 Ini karena adanya komponen integrator yang dapat menghilangkan kesalahan keadaan tunak.

4.2.2 Simulasi Pengaturan *Level Single Loop Variasi Parameter Kontroler* (K_p , τ_i , Dan τ_d)

A. *Variasi Nilai K_p*



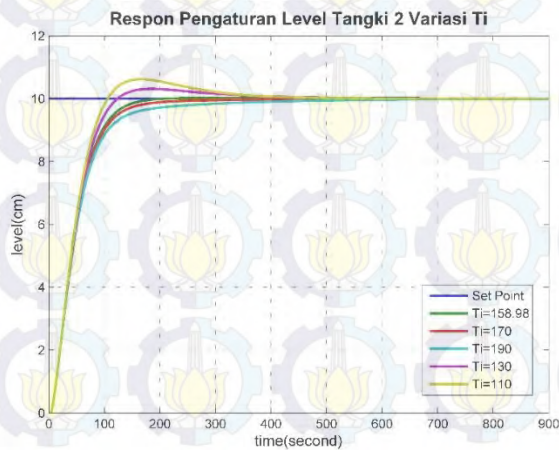
Gambar 4.4 Respon *plant* dengan variasi nilai K_p untuk sistem G_2

Pada simulasi pengujian ini dilakukan variasi nilai K_p sebanyak empat nilai yaitu untuk nilai K_p 20.61, 30, 35, 15, dan 10 dengan *set point* berupa sinyal step dengan nilai 10. Pada Gambar 4.4 ini merupakan

respon dengan variasi K_p pada pengaturan *level* pada tangki 2. Variasi nilai K_p ini tidak diikuti dengan perubahan nilai τ_i , nilai τ_i yang digunakan tetap bernilai 158.98. Pengujian terhadap perubahan nilai K_p ini bertujuan untuk membuktikan dan memahami pengaruh nilai K_p terhadap kecepatan respon dan kesalahan keadaan tunak. Dari Gambar 4.4 dapat diamati bahwa semakin besar nilai K_p maka respon dari sistem semakin cepat untuk menuju keadaan tunak, ini dapat dilihat dengan pemberian K_p sebesar 35 dibanding dengan K_p sebesar 10. Ini sesuai dengan teori bahwa K_p atau gain proporsional memiliki sifat untuk mempercepat respon. Namun nilai dari K_p juga tidak boleh terlalu besar karena akan membuat respon berosilasi dan membuat sistem tidak stabil, ini bisa disebabkan nilai *error* akan terus dikali sehingga menuju nilai yang besar.

B. Variasi Nilai τ_i

Simulasi pada pengaturan *level* dengan memberikan variasi parameter τ_i ini bertujuan untuk mengetahui pengaruh dari komponen integrator pada kontroler PID ini pada sistem atau *plant*. Sesuai dengan teori bahwa komponen integrator mampu menghilangkan kesalahan keadaan tunak pada *plant*. Simulasi pengujian ini dilakukan dengan memberikan sinyal step dengan *set point* 10 kemudian dilakukan variasi nilai τ_i .

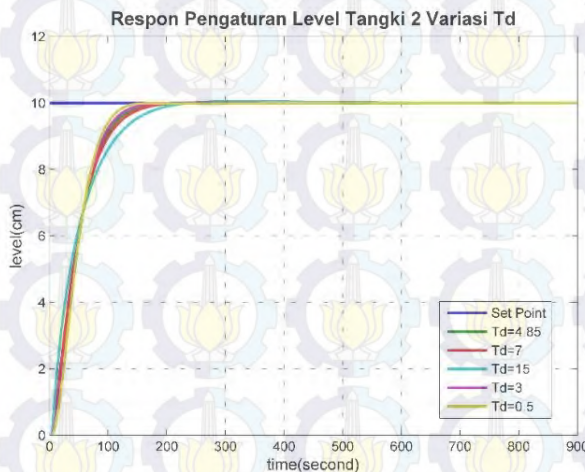


Gambar 4.5. Respon *plant* dengan variasi nilai τ_i untuk sistem G_2

Dari respon sistem pada Gambar 4.5 dapat diamati bahwa sistem memiliki kesalahan keadaan tunak 0, dan karena nilai K_p yang digunakan sama maka nilai τ dari setiap respon sama. Dapat diamati juga bahwa apabila nilai τ_i semakin kecil pada sistem akan muncul *overshoot*, dan penurunan nilai τ_i ini berbanding lurus dengan semakin besarnya *overshoot* pada sistem. Ini sesuai dengan teori bahwa semakin kecil nilai τ_i atau semakin besar nilai K_i maka *overshoot* akan semakin meningkat juga.

C. Variasi Nilai τ_d

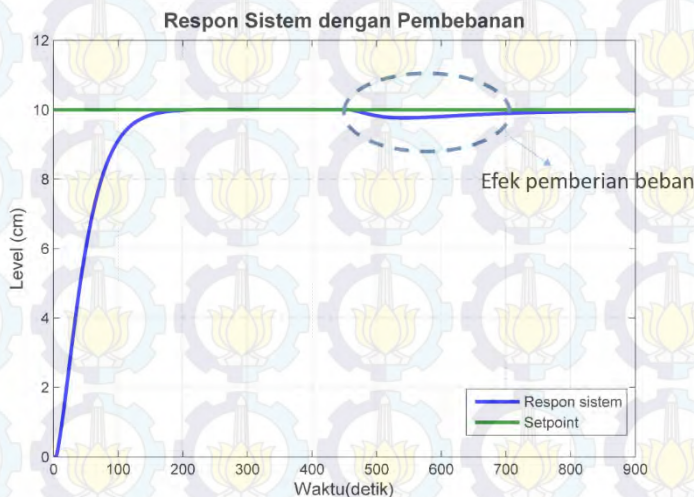
Konstanta differensial memberikan aksi berdasarkan nilai error yang akan datang. Keuntungan dari konstanta differensial adalah memiliki *zero* pada origin sehingga jika ditambahkan, sistem bisa menjadi lebih stabil seperti yang ditunjukkan pada Gambar 3.3. Pada Gambar 3.3 *locus* dari proses *level* tangki 2 dapat ditarik ke sebelah kiri sumbu imajiner. Simulasi pada *plant* dengan variasi nilai τ_d pada kontroler PID ini untuk mengetahui pengaruh dari konstanta differensial. Respon yang didapat dari pengujian variasi nilai τ_d seperti pada Gambar 4.6.



Gambar 4.6 Respon *plant* dengan variasi nilai τ_d untuk sistem G_2

Dari hasil respon dapat diamati bahwa semakin besar nilai τ_d maka respon dari sistem akan lebih lambat. Apabila respon dari *plant* memiliki *overshoot*, dengan semakin besar nilai τ_d dapat juga mereduksi *overshoot* dari respon pada *plant*, karena kecepatan respon berbanding lurus dengan *overshoot*. Semakin cepat respon maka kemungkinan *overshoot* juga semakin besar. Ini sesuai dengan sifat komponen τ_d yang dapat mereduksi *overshoot*.

4.2.3 Simulasi Sistem Pengaturan Level Single Loop Dengan Diberi Beban

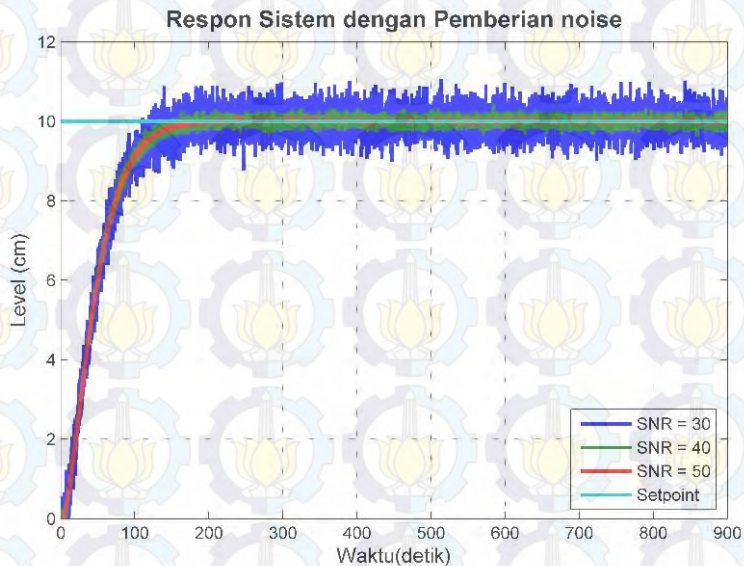


Gambar 4.7 Respon *plant* dengan pemberian beban

Pada pengaturan *level* pada tangki gangguan beban ini seperti dibukannya *valve* aliran keluar dari fluida pada saat pengisian fluida ke dalam tangki. Simulasi sistem pengaturan *level* dengan diberikannya beban ini bertujuan untuk menguji seberapa cepat kontroler mampu mengembalikan respon menuju ke kondisi stabil ketika diberi gangguan. Selain itu juga dilihat seberapa tahan sistem dengan sebuah kontroler dalam menghadapi pembebanan yang ada, dikarenakan juga pembebanan pada sistem dapat mengakibatkan perubahan parameter pada *plant*. Pada

Gambar 4.7 dapat diamati mengenai pemberian beban pada sistem. Beban diberikan pada saat respon telah mencapai kondisi tunak yaitu pada detik ke 450. Dengan adanya pembebanan terdapat penurunan nilai pada respon sistem. Untuk analisis respon dilakukan dengan perhitungan RMSE (*Root Mean Square Error*), didapat nilai RMSE dari respon sebesar 11.78%. Dari respon didapat bahwa kontroler PID tetap mampu meredam beban atau *disturbances* yang diberikan tetapi membutuhkan waktu yang cukup lama untuk mengembalikan respon ke keadaan semula yaitu dengan waktu 311.5 detik.

4.2.4 Simulasi Sistem Pengaturan *Level Single Loop* Dengan *Noise*



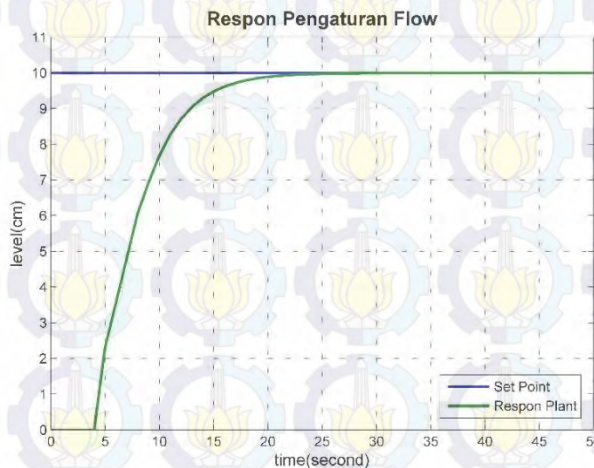
Gambar 4.8 Respon *plant* dengan diberi *noise*

Measurement noise dapat merusak informasi mengenai variabel proses yang dikirim oleh sensor yang terdapat pada *plant*. *Noise* yang diberikan pada simulasi ini merupakan *noise* AWGN. *Noise* AWGN memiliki sifat-sifat *additive*, *white*, dan *Gaussian*. Sifat *additive* artinya *noise* ini dijumlahkan dengan sinyal, sifat *white* artinya *noise* tidak bergantung pada frekuensi operasi sistem dan memiliki rapat daya yang

konstan, dan sifat *Gaussian* artinya besarnya tegangan *noise* memiliki rapat peluang terdistribusi *Gaussian*. SNR pada *noise AWGN* merupakan perbandingan (*ratio*) antara kekuatan sinyal (*signal strength*) dengan kekuatan derau (*noise level*). Pengujian *noise* ini bertujuan untuk mengukur seberapa sensitif sistem dengan sebuah kontroler terhadap perubahan kecil. Dari respon pada Gambar 4.8 ini dapat diamati bahwa respon dari sistem ini dapat tetap menuju ke *set point* atau sinyal referensi walau diberi gangguan *noise* dengan nilai SNR sebesar 30,40, 50. Nilai RMSE dari respon sistem dengan SNR 30 yaitu sebesar 31.48%, untuk SNR 40 sebesar 15.72 % dan untuk SNR sebesar 50 didapat nilai 12.55%

4.2.5 Simulasi Sistem Pengaturan Flow Single Loop

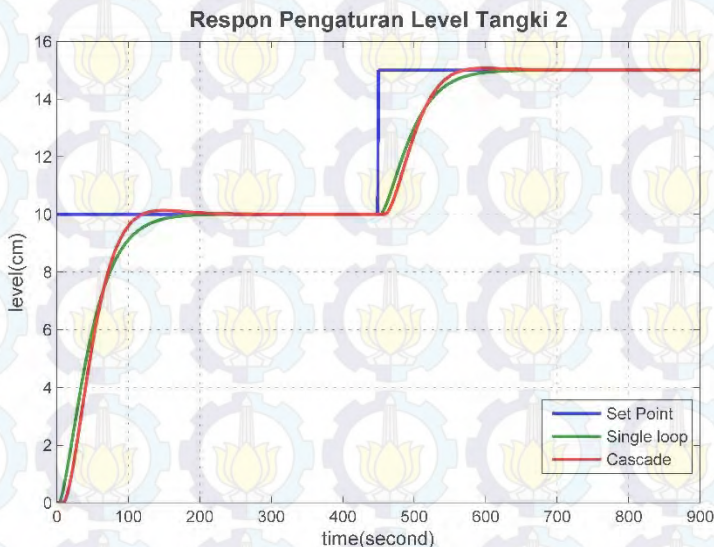
Pada respon *open loop* pengaturan *flow* didapat masih terdapat kesalahan keadaan tunak pada respon sistem, sehingga dirancang kontroler PID untuk menghilangkan kesalahan keadaan tunak tersebut. Kontroler PID dirancang dengan metode *tuning* dari Miluse Viteckova dan Antonin Vitecek dan didapatkan persamaan kontroler PID ini seperti pada Persamaan (3.59). Dengan menggunakan kontroler PID ini didapat respon dari sistem pengaturan *flow* ini seperti pada Gambar 4.9. Kesalahan keadaan tunak pada *plant* pengaturan *flow* ini dapat dihilangkan dan tidak ada *overshoot* pada respon sistem.



Gambar 4.9 Respon proses *flow* setelah diberi kontroler

4.2.6 Simulasi Sistem Pengaturan Level Konfigurasi Cascade

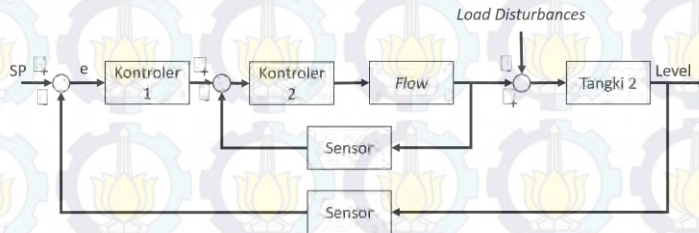
Pada tahap ini dilakukan simulasi pengaturan *level* dengan konfigurasi *cascade* dengan proses *flow*. Dari hasil simulasi didapat respon seperti pada Gambar 4.10. Respon dari pengaturan *level* dengan konfigurasi *cascade* ini langsung dibandingkan dengan respon pengaturan *level* dengan konfigurasi *single loop*. Dari hasil perhitungan RMSE pada kedua respon, didapat nilai RMSE dari pengaturan *level single loop* sebesar 0.39% sedangkan untuk pengatur *level* dengan konfigurasi *cascade* memiliki nilai RMSE sebesar 0.35%. Dari hasil perhitungan RMSE didapatkan RMSE pada respon sistem pengaturan level dengan konfigurasi *cascade* lebih baik, ini bisa disebabkan dengan dikontrolnya laju aliran air yang masuk ke tangki memiliki respon yang lebih baik yang berpengaruh terhadap level pada tangki. Dari hasil simulasi, didapat respon dari pengaturan *level* dengan konfigurasi *cascade* memiliki *overshoot* sebesar 1.35%



Gambar 4.10 Respon pengaturan *level* konfigurasi *cascade* dibandingkan dengan konfigurasi *single loop*

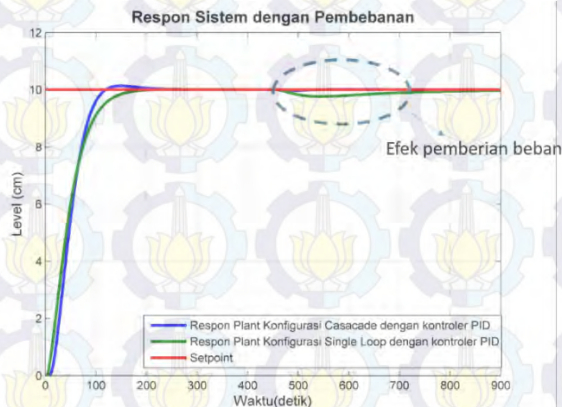
4.2.7 Simulasi Sistem Pengaturan *Level* Konfigurasi *Cascade* Dengan Diberi *Disturbance*

Pada tahap ini dilakukan simulasi pada sistem pengaturan *level* konfigurasi *cascade* dengan diberi *disturbance* dan dibandingkan hasilnya dengan pengaturan *level* dengan konfigurasi *single loop*. Diagram blok dari simulasi ini seperti pada Gambar 4.11.

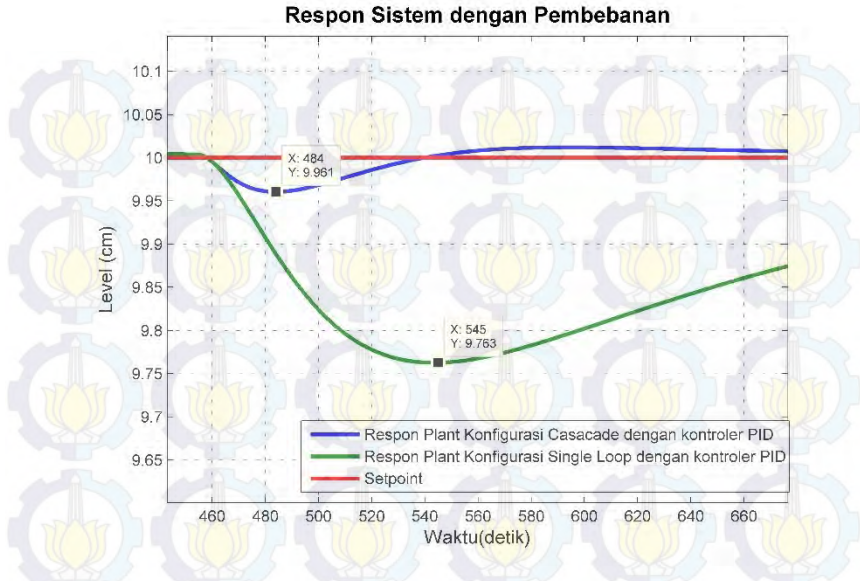


Gambar 4.11 Diagram blok sistem pengaturan konfigurasi *cascade* dengan *disturbance*

Dari hasil simulasi didapat respon seperti pada Gambar 4.12, respon dengan konfigurasi *cascade* mampu meredam *disturbance* lebih baik daripada dengan konfigurasi *single loop*. Efek pemberian beban dapat dilihat lebih jelas seperti pada Gambar 4.13.



Gambar 4.12 Respon dengan *disturbance* pada pengaturan *level* dengan konfigurasi *cascade* dan konfigurasi *single loop*



Gambar 4.13 Efek pemberian beban pada respon *plant*

Dari Gambar 4.13 pada konfigurasi *Cascade* penurunan nilai akibat pemberian beban sebesar 0.049 sedangkan untuk *single loop* sebesar 0.283. Dari perhitungan RMSE didapat bahwa untuk konfigurasi *single loop* memiliki RMSE sebesar 11.78 % dan untuk konfigurasi *Cascade* sebesar 1.13%. Ini dikarenakan pada pengaturan *cascade*, *disturbance* yang disebabkan pada *inner loop* yaitu aliran fluida yang masuk ke tangki mampu diredam terlebih dahulu dengan baik oleh adanya kontroler pada *inner loop*.

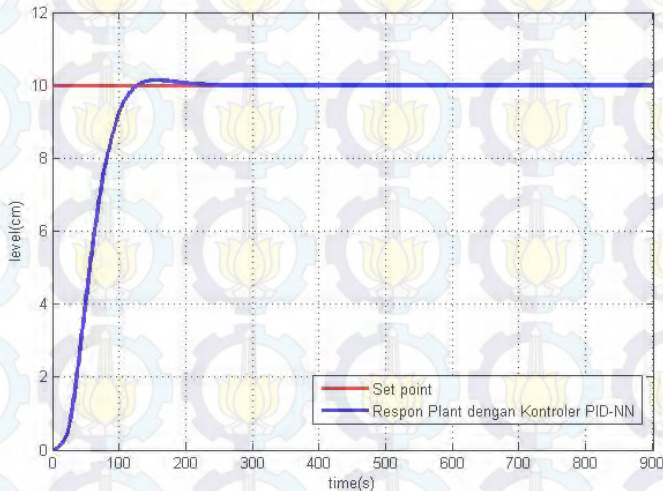
4.3 Simulasi Sistem dengan Kontroler PID *Neural-Network*

Pada tahap ini dilakukan simulasi pengaturan *level* pada *plant coupled tank* dengan menggunakan kontroler PID *Neural-Network*. Simulasi ini dilakukan dengan sinyal referensi berupa sinyal step. Simulasi yang dilakukan antara lain:

1. Simulasi sistem pengaturan *level* air pada Tangki 2
2. Simulasi dengan *learning rate*
3. Simulasi dengan pemberian beban

4.3.1 Simulasi Pengaturan *Level* Air Pada Tangki 2

Pada tahap ini dilakukan simulasi sistem pengaturan pengendalian *level* air pada tangki 2 *plant coupled tank* dengan menggunakan kontroler PID - *Neural Network* dengan konfigurasi *single-loop*.

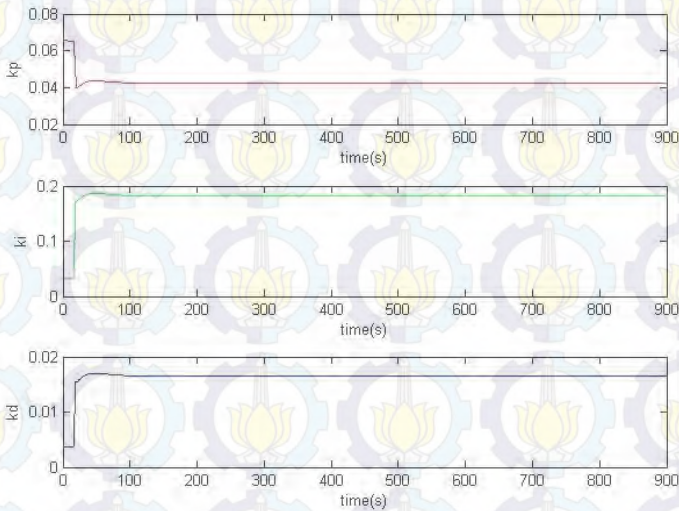


Gambar 4.14 Respon *plant* sistem pengaturan *level* dengan kontroler PID *Neural Network*

Learning rate yang digunakan pada *Neural Network* bernilai 0.9. Respon yang didapatkan dari pengujian ini seperti pada Gambar 4.14. Pada Gambar 4.14 waktu yang dibutuhkan oleh respon sistem pengaturan *level* pada tangki 2 untuk mencapai nilai 63.2% dari keadaan tunak atau 6.32 yaitu sebesar 65.98 detik. Waktu yang diperlukan ini lebih lama dibanding dengan kontroler PI.

Pada Gambar 4.15 merupakan perubahan parameter kontroler PID keluaran dari *output layer* dari *Neural Network* yang didesain. Pada Gambar 4.15 dapat diamati bahwa ketika respon dari *plant* masih pada masa transien, terjadi perubahan pada parameter kontroler PID, yang ditunjukkan dengan nilai K_p , K_i , dan K_d . Apabila respon *plant* telah memasuki keadaan tunak maka tidak terjadi perubahan pada parameter kontroler PID ini atau nilainya konstan. Dilakukan perhitungan RMSE untuk mengetahui besar *error* pada respon *plant*. Dari perhitungan RMSE

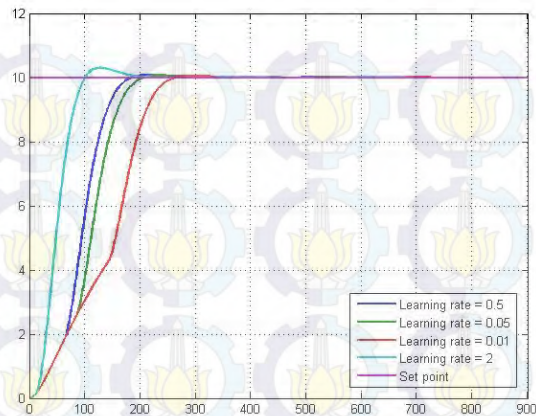
didapat nilai 0.044 %. Sehingga bisa dikatakan sistem pengaturan *level* air dengan kontroler PID- *Neural Network* mampu menghilangkan kesalahan keadaan tunak dengan baik.



Gambar 4.15 Perubahan parameter kontroler keluaran *Neural Network*

4.3.2 Simulasi Dengan Varaiasi *Learning Rate*

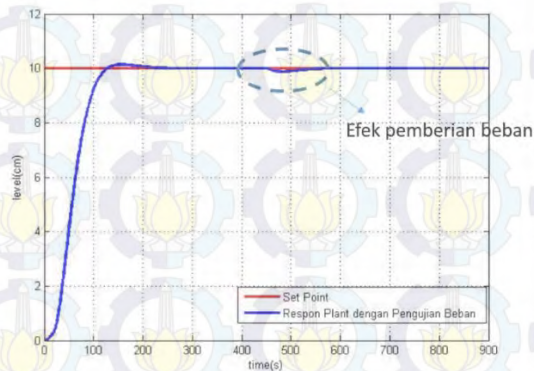
Learning rate merupakan laju pembelajaran pada algoritma *Neural Network* untuk mendapatkan nilai solusi minimum. *Learning rate* ini merupakan konstanta perhitungan pada mekanisme *backpropagation* untuk perubahan bobot pada *Neural Network*. Pada simulasi ini dilakukan pengujian dengan variasi *learning rate* bernilai 0.5, 0.05, 0.01, dan 2. Respon yang didapatkan seperti pada Gambar 4.16. Dari gambar 4.15 dapat diamati bahwa semakin besar *learning rate* maka respon *plant* semakin cepat dengan dibuktikan nilai *time constant* atau waktu yang dibutuhkan oleh respon untuk mencapai nilai 63.2% dari keadaan tunak dengan *learning rate* = 0.05 sebesar 123.8 detik, untuk *learning rate* = 0.01 sebesar 171.5 detik, dan untuk *learning rate* = 0.5 sebesar 106 detik, dan untuk *learning rate* = 2 sebesar 56.43 detik. Akan tetapi semakin besar nilai *learning rate* akan menimbulkan *overshoot* pada respon sistem.



Gambar 4.16 Respon *plant* dengan variasi *learning rate*

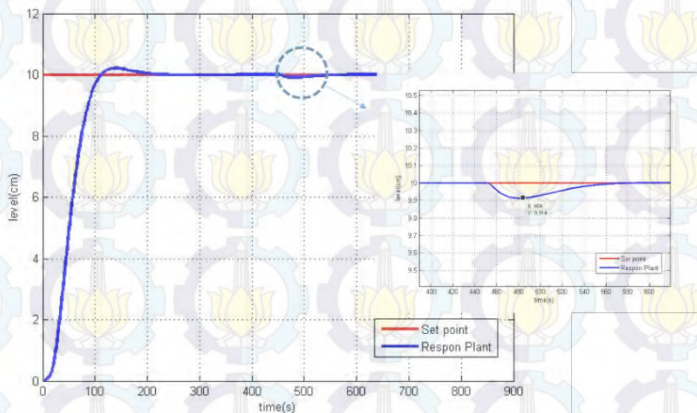
4.3.3 Simulasi Sistem *Pengaturan Level* Dengan Pemberian Beban

Dilakukan pengujian sistem dengan kontroler PID – *Neural Network* untuk konfigurasi *single loop* melalui pemberian beban ketika respon sudah mencapai tunak. Dari Gambar 4.17 dapat dilihat beban diberikan saat detik ke 450.



Gambar 4.17 Simulasi sistem pengaturan *level* dengan pemberian beban untuk konfigurasi *single loop* dengan kontroler PID-NN

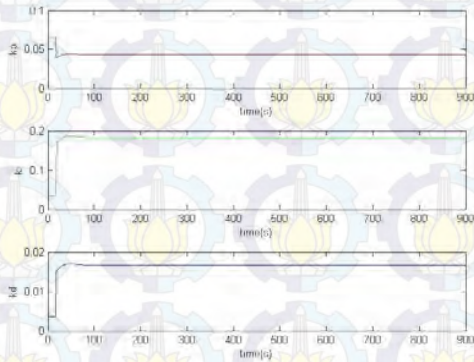
Pemberian beban disimulasikan seperti pembukaan valve saluran keluar air. Ketika sistem diberikan beban tetapi dapat kembali ke kondisi semula berarti sistem dapat dikatakan stabil. Hal tersebut dikarenakan sistem mampu beradaptasi dengan penurunan nilai yang terjadi. Penurunan nilai akibat pemberian beban pada simulasi ini sebesar 0.121. Kontroler PID – Neural Network mampu mengembalikan respon plant ke keadaan tunak dalam waktu 150 detik pada simulasi. Dari perhitungan RMSE pada pengujian beban ini didapat nilai sebesar 3,36 %. Pada pengujian dengan konfigurasi cascade didapat respon seperti pada Gambar 4.18. Dengan konfigurasi cascade didapat penurunan nilai yang lebih kecil yaitu sebesar 0.086 dan dapat kembali ke steady state dalam waktu yang lebih cepat 110 detik pada simulasi. Dari perhitungan RMSE yang dilakukan didapat nilai sebesar 5,8 %. Dari perhitungan nilai RMSE ini konfigurasi single loop memiliki nilai yang lebih baik.



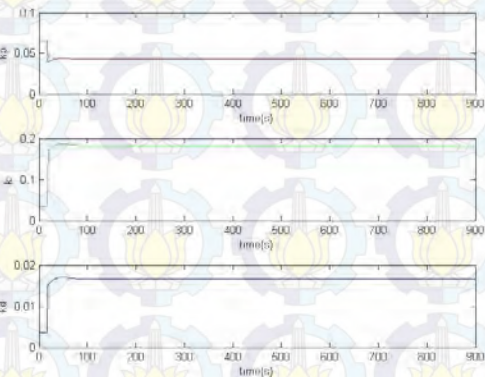
Gambar 4.18 Simulasi sistem pengaturan *level* dengan pemberian beban untuk konfigurasi *cascade* dengan kontroler PID-NN

Pada gambar 4.19 dan 4.20 dapat diamati mengenai perubahan parameter kontroler K_p , K_i , dan K_d keluaran dari *output layer* pada *Neural Network* untuk PID- *Neural Network* dengan konfigurasi *single loop* dan *cascade*. Pada waktu terjadinya pembebanan yaitu saat detik ke 450 dapat diamati bahwa tidak terjadi perubahan pada parameter kontroler. Ini bisa disebabkan sinyal kontrol yang dikeluarkan dari

kontroler PID *Neural Network* telah mencapai nilai saturasi dan kesalahan keadaan tunak yang diakibatkan oleh pembebanan ini bernilai kecil atau dalam batas toleransi nilai kesalahan yang masih bisa diperbaiki dengan parameter kontroler sebelumnya.



Gambar 4.19 Perubahan parameter kontroler keluaran *Neural Network* untuk PID *Neural Network* dengan konfigurasi *single loop*



Gambar 4.20 Perubahan parameter kontroler keluaran *Neural Network* untuk PID *Neural Network* dengan konfigurasi *cascade*



BAB 5

PENUTUP

5.1 Kesimpulan

1. Pada pengujian *set point tracking* sistem pengaturan *level plant* coupled tank didapat PID *Neural Network* memberikan nilai *error* terkecil dengan nilai RMSE 0.044%
2. Pada pengujian *disturbance* sistem pengaturan *level plant* coupled tank didapat konfigurasi *cascade* dengan kontroler PI memberikan hasil yang paling baik dengan nilai RMSE 1.13%
3. Pada pengaturan *level* tangki 2 pada *coupled tanks* dengan konfigurasi *Cascade* menggunakan kontroler PID didapat respon sistem yang memiliki *overshoot* sebesar 1.35%
4. *Learning rate* mempengaruhi kecepatan dari respon menuju tunak
5. Dengan konfigurasi *cascade* terbukti mampu meredam *disturbance* pada *plant* lebih baik daripada konfigurasi *single loop* baik menggunakan kontroler PI atau PID – *Neural Network*

5.2 Saran

Saran yang dapat diberikan penulis untuk penelitian selanjutnya adalah:

1. Pengoptimalan nilai bobot awal pada algoritma *Neural Network* sehingga bisa didapatkan hasil yang lebih baik
2. Dalam melakukan pemodelan matematis pada *plant* sebaiknya dengan mengambil data langsung dari *plant* dan kemudian dilakukan identifikasi *plant* menggunakan metode parametrik atau non parametrik

DAFTAR PUSTAKA

- [1] M.S. Ramli, M.A. Ahmad, R.M.T. Raja Ismail, "Comparison of Swarm Adaptive *Neural Network* Control of a *Coupled Tank Liquid Level System*, *International Conference on Computer Technology and Development*, Kinabalu, 13-15 November 2009.
- [2] Hector Batisda, Pedro Ponce, Ricardo Ramirez, and Arturo Molina, "Model and Control for *Coupled tanks* using Labview", *International Conference on Mechatronics, Electronics and Automotive Engineering*. Morelos, 19-22 November 2013.
- [3] Sigeru Omatu, Yoshiku Kishida, and Michifumi Yoshioka, "Neuro Control for Single-Masukan Multi Keluaran Systems", *Second International Conference on Knowledge-Based Intelligent Electronic Systems*, Adelaide, 21-23 April 1998.
- [4] Hua Ji and Zhiyong Li, "Design of *Neural Network* PID Controller Based on Brushless DC Motor", *Second International Conferenve on Intelligent Computation Technology and Automation*. Changsha, Hunan, 10-11 October 2009.
- [5] Yun Li, H. A. Kiam, dan Gregory Chong., "PID Control System Analysis, Design, and Technology. Problems, Remedies, and Future Directions" *IEEE CONTROL SYSTEM MAGAZINE*, February 2006.
- [6] S. Omatu, M. Yoshioka, T. Kosaka, H. Yanagimoto dan J. A. Dargham, "Neuro-PID Control of Speed and Torque of Electric Vehicle," *Advances Engineering Computing and Application in Sciences*, Sliema, Malta, 11-16 October 2009.
- [7] Haykin, Simon. "*Neural Networks* and Learning Machines: Third Edition". United States of America. 1999. Pearson Prentice Hall.
- [8] Ogata, Katsuhiko, "*Modern Control Engineering*", Prentice Hall, New Jersey, 1970
- [9] Yang Song, Shenyi Wu, Yuying Yan. "Development of Self-Tuning Intelligent PID Controller Based on BPNN for Indoor Air Quality Control". *International Journal of Emerging Technology and Advanced Engineering*, Volume 3, Issue 11, November 2013.

LAMPIRAN

Lampiran 1. Program Autotuning PID – Neural Network

Backpropagation

```
clear all;
clc
close all;
xite=0.9;
alfa=0.05;
S=1; %Signal type
IN=4;H=5;Out=3; %NN Structure
S=1; %Step Signal
wi=[-0.6394    -0.2696    -0.3756    -0.7023;
    -0.8603    -0.2013    -0.5024    -0.2596;
    -1.0749     0.5543    -1.6820    -0.5437;
    -0.3625    -0.0724    -0.6463    -0.2859;
     0.1425     0.0279    -0.5406    -0.7660];
%wi=0.50*randi(H,IN);
wi_1=wi;wi_2=wi;wi_3=wi;
wo=[0.7576    0.2616    0.5820    -0.1416    -
    0.1325;
    -0.1146    0.2949    0.8352    0.2205
    0.4508;
     0.7201    0.4566    0.7672    0.4962
    0.3632];
%wo=0.50*randi(Out,H);
wo_1=wo;wo_2=wo;wo_3=wo;

x=[0,0,0];
u_1=0.0;u_2=0.0;u_3=0.0;u_4=0;u_5=0;
y_1=0.0;y_2=0.0;y_3=0.0;y_4=0;y_5=0;
Oh=zeros(H,1); %Output from NN middle layer
I=Oh; %Input to NN middle layer
error_2=0;
error_1=0;
ts=10;
t=0:1:900;
for k=1:1:901
    time(k)=k;
```

```

if S==1
    rin(k)=10.0;
end

%Plant model
num=[0.157];
den=[2037 169 1];
sys=tf(num,den,'inputdelay',4.23);
dsys=c2d(sys,ts,'z');
[num,den]=tfdata(dsys,'v');
yout(k)=-den(2)*y_1-den(3)*y_2+num(2)*u_2+
num(3)*u_3 ; % tao/T=n,n>1
error(k)=rin(k)-yout(k);
xi=[rin(k),yout(k),error(k),1];
x(1)=error(k)-error_1;
x(2)=error(k);
x(3)=error(k)-2*error_1+error_2;
epid=[x(1);x(2);x(3)];
I=xi*wi';
for j=1:1:H
    Oh(j)=(exp(I(j))-exp(-
I(j)))/(exp(I(j))+exp(-I(j))); %Middle Layer
end
K=wo*Oh; %Output Layer
for l=1:1:Out
    K(l)=exp(K(l))/(exp(K(l))+exp(-K(l)));
%Getting kp,ki,kd
end
kp(k)=K(1);ki(k)=K(2);kd(k)=K(3);
Kpid=[kp(k),ki(k),kd(k)];
du(k)=Kpid*epid;
u(k)=u_1+du(k);
if u(k)>=100 % Restricting the output of
controller
    u(k)=100;
end
if u(k)<=-100
    u(k)=-100;
end

```

```

dyu(k)=sign((yout(k)-y_1)/(u(k)-u_1));
E(k)=0.5*(rin(k)-yout(k))^2;
%Output layer
for j=1:1:Out
    dK(j)=K(j)*(1-K(j));
end
for l=1:1:Out
    delta3(l)=error(k)*dyu(k)*epid(l)*dK(l);
end
for l=1:1:Out
    for i=1:1:H
        d_wo=xite*delta3(l)*Oh(i)+alfa*(wo_1-
wo_2);
    end
end
    wo=wo_1+d_wo;
%Hidden layer
for i=1:1:H
    dO(i)=(1-(Oh(i))^2)/2;
end
    segma=delta3*wo;
for i=1:1:H
    delta2(i)=dO(i)*segma(i);
end
d_wi=xite*delta2'*xi;
wi=wi_1+d_wi;
%Parameters Update
u_5=u_4;u_4=u_3;u_3=u_2;u_2=u_1;u_1=u(k);
y_5=y_4;y_4=y_3;y_3=y_2;y_2=y_1;y_1=yout(k);
wo_3=wo_2;
wo_2=wo_1;
wo_1=wo;
wi_3=wi_2;
wi_2=wi_1;
wi_1=wi;
error_2=error_1;
error_1=error(k);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

Kp=23;
Ti=163.93;
s=tf('s');
C=tf(Kp*(1+1/(Ti*s)));
P=feedback(C*sys,1);
B1=lsim(P,rin,t);
%
P1=tf([0.92],[0.392 1],'inputdelay',4.23);
Kp1=0.036;
Ti1=0.392;
C1=tf(Kp1*(1+1/(Ti1*s)));
T1=feedback(C1*P1,1);
P1=feedback(C*T1*sys,1);
B2=lsim(P1,rin,t);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%bpid_s_plot
figure(1);
plot(time,rin,'r',time,yout,'b',time,B1,'g','linewidth',2.5);
xlabel('time(s)');ylabel('level(cm)');
figure(2);
plot(time,error,'r');
xlabel('time(s)');ylabel('error');
figure(3);
plot(time,u,'r');
xlabel('time(s)');ylabel('u');
figure(4);
subplot(311);
plot(time,kp,'r');
xlabel('time(s)');ylabel('kp');
subplot(312);
plot(time,ki,'g');
xlabel('time(s)');ylabel('ki');
subplot(313);
plot(time,kd,'b');
xlabel('time(s)');
ylabel('kd');
figure(5)

```

```
plot(time, rin, 'r', time, yout, 'b', time, B2, 'g', 'lin  
ewidth', 2.5);  
xlabel('time(s)'); ylabel('level(cm)');  
figure(6)  
plot(time, rin, 'r', time, yout, 'b', 'linewidth', 2.5)  
;  
xlabel('time(s)'); ylabel('level(cm)');
```

Lampiran 2. Program Matlab Sistem Pengaturan Cascade Level pada Tangki 2

```
%Regulate Level in Tank 2
t=0:1:900 ;% sampling time 0.1 s dengan jumlah
data 1000
for i = 1:451;
    p(i)=10;
for i=451:901;
    p(i)=15;

end
end
for i = 1:901;
    q(i)=10;
end
for i = 1:901;
    r(i)=5;
end
% Tf H2/Q
P1=tf([0.157],[2037 169 1],'inputdelay',4.23);
% Tf flow
P2=tf([0.92],[0.392 1],'inputdelay',4.23);
%Controller PI for plant P1
Kp1=25;
ti1=163.93;
s=tf('s');
C1=tf(Kp1*(1+1/(ti1*s)));
%Controller PI for plant Flow
Kp2=0.036;
ti2=0.392;
C2=tf([Kp2*ti2 Kp2],[ti2 0]);
%Feedback H1/Q
A=feedback(C1*P1,1);
%Feedback flow
E=feedback(C2*P2,1);
%Cascade H2 and flow
F=feedback(E*C1*P1,1);
%Output
A1=lsim(A,p,t); %single loop
```

```

B1=lsim(F,p,t); %cascade
B2=10*step(A,t); %single loop
A2=10*step(F,t); %cascade
figure(1)
plot(t,p,t,B1,t,A1,'linewidth',2),grid on
title('Respon Pengaturan Level Tangki
2','FontWeight','Bold','FontSize',14)
xlabel('time(second)','FontWeight','normal','Font
tSize',12)
ylabel('level(cm)','FontWeight','normal','FontSi
ze',12)
legend('Set Point','Single loop','Cascade')
figure(2)
plot(t,q,t,B2,t,A2,'linewidth',2),grid on
title('Respon Pengaturan Level Tangki
2','FontWeight','Bold','FontSize',14)
xlabel('time(second)','FontWeight','normal','Font
tSize',12)
ylabel('level(cm)','FontWeight','normal','FontSi
ze',12)
legend('Set Point','Single loop','Cascade')
figure(3)
plot(t,q,t,A2,'linewidth',2),grid on
title('Respon Pengaturan Level Tangki
2','FontWeight','Bold','FontSize',14)
xlabel('time(second)','FontWeight','normal','Font
tSize',12)
ylabel('level(cm)','FontWeight','normal','FontSi
ze',12)
legend('Set Point','Single loop','Cascade')

```


Lampiran 3. Program Sistem Pengaturan Level pada Tangki 2 dengan diberi dsiturbance

```
clear all
clc
m=0;
%% Inisialisasi waktu
t=0:0.1:900;           %Waktu Simulasi dan banyaknya
data yang diambil
%% Inisialisasi Masukan
Masukan(1,1:9001)=10;    %Masukan berupa
sinyal unit step
Masukan1(1,1:701)=10;    %Masukan saat
pembebanan 1
Masukan2(1,1:651)=10;    %Masukan saat
pembebanan 2
%% Penyusunan fungsi alih sistem
%tanpa beban
P=tf([0.157],[2037 169 1],'inputdelay',4.23);
%fungsi alih plant
s=tf('s');
Kp=23;
Ti=163.93;
C=tf(Kp*(1+1/(Ti*s)));    %fungsi alih
kontroler PI
T = feedback(C*P,1);      %fungsi alih
loop tertutup

%Beban
D=[0.75];
B=feedback(P,C);

%Pemberian noise
n2a=awgn(Masukan,20,'Measured');n2a=n2a-Masukan;
n2b=awgn(Masukan,30,'Measured');n2b=n2b-Masukan;
n2c=awgn(Masukan,40,'Measured');n2c=n2c-Masukan;
T3 = feedback(1,C*P);

%% Simulasi
%Pembebanan
```

```

for i = 1:9001;
%Inisialisasi sinyal
    pi(i)=50;
    as(i)=1;
%Noise signal
end
for n = 1:1
    for i = 1:4500;
%Inisialisasi sinyal
        sda(i,n)=0;
%Noise signal
    end
    for i = 4501:9001;
%Inisialisasi sinyal
        sda(i,n)=D(n);
%Noise signal
    end
    for l = 1:1
m=m+1;
RP(:,n) = lsim(T,as,t);
Load=P/(1+P*C);
SL(:,n)=lsim(Load,sda(:,n),t);
RnB(:,m) = (-SL(:,m)+RP(:,m))*10;
    end
end
%Pemberian noise
B21(:,1) = (10*step(T,t))+lsim(T3,n2a,t);
B22(:,1) = (10*step(T,t))+lsim(T3,n2b,t);
B23(:,1) = (10*step(T,t))+lsim(T3,n2c,t);

%% Plotting Hasil
figure (1)
plot(t,RnB,t,Masukan,'linewidth',2.5),grid on
title('Respon Sistem dengan
Pembebanan','FontWeight','Bold','FontSize',14)
ylabel('Level
(cm)','FontWeight','normal','FontSize',12)
xlabel('Waktu(detik)','FontWeight','normal','FontSize',12)

```

```
%legend('Respon sistem','Setpoint')
```

```
figure (2)  
plot(t,B21,t,B22,t,B23,t,Masukan,'linewidth',2),  
grid on  
title('Respon Sistem dengan Pemberian  
noise','FontWeight','Bold','FontSize',14)  
ylabel('Level  
(cm)','FontWeight','normal','FontSize',12)  
xlabel('Waktu(detik)','FontWeight','normal','FontSize',12)  
legend('SNR = 20','SNR = 30','SNR =  
40','Setpoint')
```

RIWAYAT PENULIS



Rhezha Qashmal Darmawan, 5 Juni 1994. Merupakan anak pertama dari 2 bersaudara pasangan ibu Prihat Assih dan bapak Agus Dwi Swandaru. Penulis menamatkan pendidikan Sekolah Dasar di SDI Sabilillah Malang, SMPN 3 Malang, dan SMAN 3 Malang. Setelah menamatkan jenjang SMA, penulis melanjutkan kuliah S1 Teknik Elektro di Institut Teknologi Sepuluh Nopember melalui jalur Program Kemitraan dan Mandiri ITS. Penulis mengambil konsentrasi bidang studi Teknik Sistem Pengaturan. Pada bulan Januari, penulis mengikuti ujian Tugas Akhir sebagai syarat untuk menyelesaikan masa studi dan memperoleh gelar Sarjana Teknik Elektro.