



TUGAS AKHIR - TF 141581

**PERANCANGAN *CASCADE CONTROL*  
MENGUNAKAN *POSITIONER* SEBAGAI  
*SECONDARY CONTROLLER***

Muhamad Muiz Hernawan  
NRP.2415105010

Dosen Pembimbing  
Totok Ruki Biyanto, Ph.D

DEPARTEMEN TEKNIK FISIKA  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



**FINAL PROJECT - TF141581**  
**DESIGN OF CASCADE CONTROL USING**  
**POSITIONER AS SECONDARY CONTROLLER**

Muhamad Muiz Hernawan  
NRP.2415105010

*Supervisor*  
Totok Ruki Biyanto, Ph.D

*DEPARTMENT OF ENGINEERING PHYSICS*  
*Faculty of Industrial Technology*  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017

## PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Muhamad Muiz Hernawan

NRP : 2415105010

Departemen : Teknik Fisika FTI-ITS

Dengan ini menyatakan bahwa tugas akhir saya yang berjudul “PERANCANGAN *CASCADE CONTROL* MENGGUNAKAN *POSITIONER* SEBAGAI *SECONDARY CONTROLLER*” adalah bebas plagiasi. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 1 Agustus 2017  
Yang membuat pernyataan,



Muhamad Muiz Hernawan

**LEMBAR PENGESAHAN**  
**PERANCANGAN *CASCADE CONTROL***  
**MENGGUNAKAN *POSITIONER* SEBAGAI**  
***SECONDARY CONTROLLER***

**Oleh:**

**Muhamad Muiz Hernawan**  
**NRP. 2415 105 010**

**Surabaya, 1 Agustus 2017**  
**Mengetahui/Menyetujui**

**Pembimbing**



**- Totok Ruki Biyanto, Ph.D**  
**NIP. 19710702 199802 1 001**



**PERANCANGAN CASCADE CONTROL  
MENGUNAKAN POSITIONER SEBAGAI  
SECONDARY CONTROLLER**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
pada  
Bidang Studi Instrumentasi  
Program Studi S-I Departemen Teknik Fisika  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember

Oleh :

**MUHAMAD MUIZ HERNAWAN**  
**NRP. 2415 105 010**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Totok Ruki Biyanto, Ph.D ..... (Pembimbing)
2. Dr. Ir. Ali Musyafa, MSc ..... (Ketua Penguji)
3. Dr. Ing. Doty Dewi Risanti, ST, MT ..... (Penguji I)
4. Dr. Gunawan Nugroho, ST, MT ..... (Penguji II)

**SURABAYA**  
**JULI 2017**

*Halaman ini sengaja dikosongkan*

# PERANCANGAN *CASCADE CONTROL* MENGUNAKAN *POSITIONER* SEBAGAI *SECONDARY CONTROLLER*

Nama Mahasiswa : Muhamad Muiz Hernawan  
NRP : 2415105010  
Departemen : S1 Teknik Fisika  
Dosen Pembimbing : Totok Ruki Biyanto Ph.D

## Abstrak

Laporan ini mengusulkan tentang metode pengendalian baru yaitu *cascade control sensorless*. *Cascade control* adalah pengendalian dimana terdapat 2 *controller* dan 2 *transmitter*. Metode kontrol tersebut digunakan pada sebuah *plant* yang memerlukan respon yang cepat. *Smart positioner* merupakan *controller* pada *control valve* yang digunakan untuk menjaga performansi dari *control valve*. Kegunaan dari *positioner* yang sebenarnya masih dapat dimaksimalkan salah satunya adalah kontrol PID yang terdapat pada *positioner*. Penelitian ini memanipulasi nilai KP pada *level controller* dan *smart positioner*, dengan cara memanfaatkan nilai *gain flow* sebagai pembagi nilai KP dari *smart positioner*. *Cascade control sensorless* yang dirancang memanfaatkan *smart positioner* sebagai *secondary controller* pada *cascade control* dengan *gain* sebesar 0,0284. Hasil perbedaan respon yang didapat *cascade control* dan *cascade control sensorless* pada *max overshoot* sebesar 0,071% dan *settling time* sebesar 50 detik sehingga *cascade control sensorless* dapat menggantikan *cascade control* pada *real plant*. Keunggulan *cascade control sensorless* apabila dibandingkan dengan *cascade control* hanya menggunakan 1 *transmitter* dan 1 *controller*, dimana peran *secondary control* sudah digantikan dengan *smart positioner* pada *control valve*.

**Kata Kunci** – Pengendalian, *Cascade Control*, *Positioner*

*Halaman ini sengaja dikosongkan*



## **DESIGN OF CASCADE CONTROL USING POSITIONER AS SECONDARY CONTROLLER**

**Name** : Muhamad Muiz H  
**NRP** : 2415105010  
**Department** : S1 Engineering Physics  
**Supervisor** : Totok Ruki Biyanto Ph.D

### **Abstract**

*This report proposes a new control method of sensorless cascade control. Cascade control involves 2 controllers and 2 transmitters, for a plant that requires a fast response. Smart positioner is a controller on the control valve which is used to maintain the performance of the control valve. The benefits of the actual positioner can still be maximized, one of them is by manipulating the PID control located in the positioner. This research manipulated values of P at the level controller with gain 0,0284 and smart positioner, Sensorless cascade controls are designed utilizing the smart positioner as a secondary controller in the cascade control. Research shows that the responses obtained by cascade control and sensorless cascade control differ only about 0,071% for max overshoot and 50 second for settling time so that the sensorless cascade control can replace the cascade control at the real plant. The advantage of cascade control sensorless is it uses only 1 transmitter and 1 controller, where the role of secondary control has been replaced with smart positioner on the control valve.*

**Keywords - Control, Cascade Control, Positioner**

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah S.W.T, karena rahmat dan hikmat-Nya sehingga penulis diberikan kesehatan, kemudahan, dan kelancaran dalam menyusun laporan tugas akhir ini. Tidak lupa juga penulis menyampaikan ucapan terima kasih kepada keluarga dan para sahabat, atas dukungan mereka, penulis mampu menyusun laporan tugas akhir ini. Tugas akhir ini merupakan salah satu persyaratan akademik yang harus dipenuhi dalam Program Studi S-1 Teknik Fisika FTI-ITS. Penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Agus Muhamad Hatta, ST, MSi, Ph.D. selaku ketua departemen Teknik Fisika ITS.
2. Segenap keluarga yang telah memberikan dukungan penuh dalam penyelesaian tugas akhir ini.
3. Bapak Totok Ruki Biyanto Ph.D selaku dosen pembimbing tugas akhir ini, yang selalu sabar memberikan semangat dan masukan-masukan kepada penulis sehingga penulis dapat menyelesaikan tugas akhir.
4. Segenap Bapak/Ibu dosen pengajar di Departemen Teknik Fisika - ITS.
5. Rekan-rekan LJ GENAP TF 2015 dan warga Teknik Fisika - ITS, yang senantiasa memberikan motivasi dan perhatian.
6. Teman-teman seperjuangan TA yang telah memotivasi dan memberikan bantuan-bantuan dalam penyelesaian laporan tugas akhir ini.

Penulis menyadari bahwa mungkin masih ada kekurangan dalam laporan ini, sehingga kritik dan saran penulis terima. Semoga laporan ini dapat berguna dan bermanfaat bagi penulis dan pihak yang membacanya.

Surabaya, 2017

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PERNYATAAN BEBAS PLAGIASI</b> Error! Bookmark not defined.	
<b>LEMBAR PENGESAHAN</b> .....	iii
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	ix
<b>KATA PENGANTAR</b> .....	xi
<b>DAFTAR ISI</b> .....	xii
<b>DAFTAR GAMBAR</b> .....	xiv
<b>DAFTAR TABEL</b> .....	xvi
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah .....	2
1.3 Tujuan Tugas Akhir .....	2
1.4 Batasan Masalah.....	2
<b>BAB II TINJAUAN PUSTAKA</b> .....	3
2.1 <i>Control Valve</i> .....	3
2.2 <i>Positioner</i> .....	4
2.3 <i>Close Loop</i> .....	5
2.4 <i>Cascade control</i> .....	6
2.5 <i>Controller PID (Proportional Intregal Derivative)</i> .....	8
<b>BAB III METODOLOGI PENELITIAN</b> .....	11
3.1 Skema Diagram Alir Penelitian .....	11
3.2 Perancangan <i>plant</i> .....	12
3.3 Pemodelan Control Valve .....	15
3.4 Perancangan <i>Control PID</i> .....	16
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	23
4.1 Perhitungan Control Valve .....	23
4.2 Pemodelan Plant dan <i>Control Valve</i> .....	24
4.3 Hasil Simulink Close Loop Level.....	29
4.4 Hasil Simulink <i>Close Loop Level</i> Dengan Konstanta .....	31
4.5 Hasil Pemodelan Cascade Control Sensorless.....	33

4.6 Perbandingan Hasil Respon Simulink.....	35
4.7 Perbandingan Hasil Respon <i>Real Plant</i> .....	36
<b>BAB V PENUTUP</b> .....	41
5.1 Kesimpulan.....	41
5.2 Saran.....	41
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

<b>Gambar 2. 1</b>	Aktuator control valve.....	3
<b>Gambar 2. 2</b>	Body control valve .....	4
<b>Gambar 2. 3</b>	Positioner .....	5
<b>Gambar 2. 4</b>	Diagram blok close loop.....	6
<b>Gambar 2. 5</b>	Skema cascade control .....	7
<b>Gambar 2. 6</b>	Diagram blok pengendalian proportional .....	9
<b>Gambar 2. 7</b>	Diagram blok pengendalian integral .....	9
<b>Gambar 3. 1</b>	<i>Flowchart</i> pengerjaan tugas akhir .....	11
<b>Gambar 3. 2</b>	Diagram blok <i>close loop real plant</i> .....	13
<b>Gambar 3. 3</b>	Proses <i>flow diagram plant</i> .....	13
<b>Gambar 3. 4</b>	<i>Plant level real plant</i> .....	14
<b>Gambar 3. 5</b>	Diagram blok <i>Cascade control</i> .....	16
<b>Gambar 3. 6</b>	Diagram blok <i>Cascade control real plant</i> .....	16
<b>Gambar 3. 7</b>	<i>P&amp;ID cascade control real plant</i> .....	17
<b>Gambar 3. 8</b>	Diagram blok <i>close loop level real plant</i> .....	18
<b>Gambar 3. 9</b>	<i>P&amp;ID close loop level real plant</i> .....	18
<b>Gambar 3.10</b>	Diagram blok penambahan <i>feedback gain flow</i> ...	19
<b>Gambar 3.11</b>	Diagram blok <i>cascade control sensorless</i> .....	20
<b>Gambar 3.12</b>	<i>P&amp;ID cascade control sensorless level real plant</i> .....	20
<b>Gambar 4. 1</b>	Uji <i>open loop level</i> .....	26
<b>Gambar 4. 2</b>	Hasil respon <i>simulink cascade control</i> .....	27
<b>Gambar 4. 3</b>	Pemodelan <i>cascade control real plant</i> .....	28
<b>Gambar 4. 4</b>	Hasil respon <i>simulink close loop level control</i> .....	29
<b>Gambar 4. 5</b>	Pemodelan <i>close loop level</i> .....	30
<b>Gambar 4. 6</b>	Hasil respon simulink <i>close loop level</i> dengan konstanta.....	31
<b>Gambar 4. 7</b>	Pemodelan <i>close loop level</i> dengan konstanta .....	32
<b>Gambar 4. 8</b>	Hasil respon <i>simulink cascade control sensorless</i> .....	33
<b>Gambar 4. 9</b>	Pemodelan <i>cascade control sensorless</i> .....	34
<b>Gambar 4.10</b>	Perbandingan hasil respon <i>simulink close loop level, cascade control, dan cascade control sensorless</i> .....	35

<b>Gambar 4.11</b> Respon <i>close loop level</i> saat dirubah <i>setpoint</i> dari 50 cm dan 55 cm.....	37
<b>Gambar 4.12</b> Respon <i>close loop level</i> saat dirubah <i>setpoint</i> dari 50 cm dan 55 cm.....	37
<b>Gambar 4.13</b> Hasil respon <i>close loop level</i> dengan diberi disturbance .....	39
<b>Gambar 4.14</b> Hasil respon <i>cascade control sensorless</i> dengan diberi <i>disturbance</i> .....	40

## DAFTAR TABEL

<b>Tabel 3. 1</b> Nilai KP dan KI pada cascade control dari autotune Matlab.....	17
<b>Tabel 3. 2</b> Nilai KP dan KI pada close loop level dari autotune Matlab.....	19
<b>Tabel 3. 3</b> Nilai KP dan KI pada close loop level dengan penambahan gain flow dari autotune Matlab .....	19
<b>Tabel 3. 4</b> Nilai KP dan KI pada cascade control sensorless dari autotune Matlab.....	21
<b>Tabel 4. 1</b> Hasil pengambilan data dan perhitungan pada cv .....	23
<b>Tabel 4. 2</b> Hasil perhitungan mass flow rate .....	23
<b>Tabel 4. 3</b> Hasil pengambilan data time constant .....	24
<b>Tabel 4. 4</b> Hasil pengambilan data pada I/P .....	24
<b>Tabel 4. 5</b> Hasil pengambilan data pada aktuator .....	25
<b>Tabel 4. 6</b> Hasil pengambilan data pada flow.....	26
<b>Tabel 4. 7</b> Perbandingan nilai grafik .....	36
<b>Tabel 4. 8</b> Perbandingan nilai grafik real plant.....	38
<b>Tabel 4. 9</b> Perbandingan nilai grafik real plant.....	40



*Halaman ini sengaja dikosongkan*



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Industri proses banyak membutuhkan *loop* kontrol untuk memastikan proses berjalan dalam kondisi operasi yang diinginkan. Performansi yang baik dari *loop* kontrol tersebut dapat membuat kualitas produk meningkat dan efisien, sehingga *loop* kontrol yang mempunyai performansi yang baik sangat diperlukan. Performansi yang buruk seperti osilasi tinggi akan menurunkan kualitas produk dan meningkatkan biaya operasional[1]. Ada 2 cara untuk memperbaiki performansi buruk pada *loop* kontrol yaitu dengan cara mengubah metode kontrol atau melakukan *tunning* ulang pada kontrol *Proportional Integral Derivative* (PID).

*Proportional integral* (PI) *controller* sudah banyak digunakan di proses industri dikarenakan kemudahannya, ketahanan dan range yang lebar untuk diaplikasikan pada proses kontrol walaupun tidak semua *real plant* menggunakan PI karena beberapa menggunakan PID *controller*. Survey dari Jepang menunjukkan bahwa PID kontrol, *conventional advanced control* dan *model predictive control* adalah 100:10:1. Sebagai tambahan, kebanyakan PID kontrol tidak menggunakan *derivative* dikarenakan apabila respon semakin cepat maka semakin tinggi pula *overshoot* yang didapat. Walaupun PI hanya menggunakan dua parameter, ini tidak semudah untuk menemukan nilai yang terbaik dan banyak *controller* yang -tuningnya masih buruk dikarenakan pengambilan nilai *tunning* dengan menggunakan metode *trial and error*[2].

Peralatan instrumen saat ini banyak berbasis *microprocessor* khususnya di *control valve* bernama *smart positioner*. *Positioner* adalah komponen kontrol pada *control valve*, dimana secara mekanik *positioner* terhubung pada aktuator dan nantinya akan menggerakkan aktuator berdasarkan masukan dari *controller*. *Smart positioner* sendiri

mempunyai keunggulan *auto calibrate* dan dapat difungsikan sebagai *controller* PI. Kontrol PID pada *positioner* berfungsi sebagai penjaga presisi bukaan dari *control valve*, apabila dilakukan *tuning* yang tidak benar maka akan berpengaruh pada kerja *control valve*[3]. Apabila ada *disturbance* yang terjadi pada *plant* maka akan mempengaruhi proses, jika tidak dilakukan *tuning* pada *positioner* maka proses akan menjadi lambat. Nilai *tuning* pada *positioner* yang tepat dapat mempercepat proses apabila ada *disturbance*.

### 1.2 Perumusan Masalah

Dari latar belakang yang telah diuraikan, maka permasalahan yang diangkat pada penelitian ini adalah sebagai berikut:

- a. Perancangan *Smart Positioner* yang bisa menggantikan kontrol *flow* yang *sensorless*.
- b. Hasil kinerja sistem yang dikerjakan.

### 1.3 Tujuan Tugas Akhir

Tujuan dari penelitian ini berdasarkan permasalahan yang diuraikan, adalah sebagai berikut ini:

- a. Merancang *Smart Positioner* yang bisa menggantikan kontrol *flow* yang *sensorless*.
- b. Menganalisa hasil kinerja sistem yang dikerjakan

### 1.4 Batasan Masalah

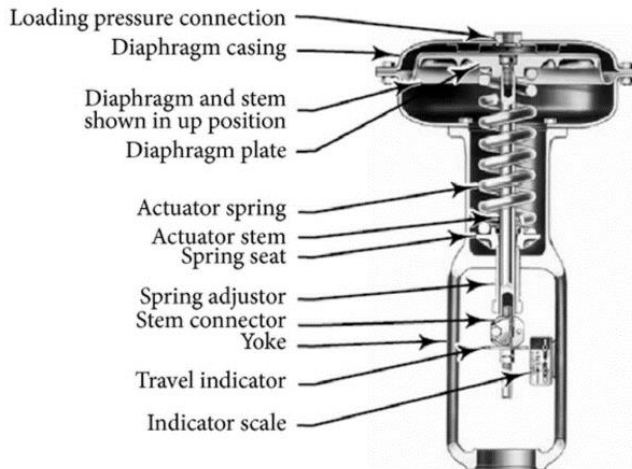
Batasan masalah dalam tugas akhir ini antara lain :

- a. Pengambilan data dilakukan dengan cara pembacaan data primer
- b. Pemodelan *control valve* didapatkan dari data *output pressure, open percentage stroke, flow rate*.
- c. Perancangan kontrol PID dengan menggunakan *software* Matlab dengan cara *auto tune*
- d. Pengaplikasian PI pada *smart positioner*

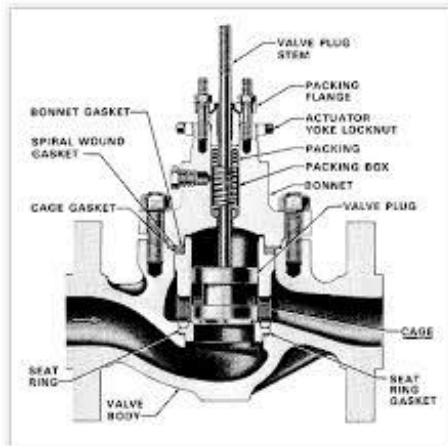
## BAB II TINJAUAN PUSTAKA

### 2.1 Control Valve

*Control Valve* merupakan *final control element* dalam sebuah sistem pengendalian. Fungsi dari *control valve* secara umum adalah untuk mengatur besar kecilnya fluida yang akan dialirkan menuju proses selanjutnya. Tipe *control valve* berdasarkan arah *supply* yang masuk terdiri dari 2 *control valve* yaitu *fail to open* atau *air to close*, *fail to close* atau *air to open*. *Air to open* adalah tipe *control valve* yang bergerak atau membuka ketika ada udara yang masuk, sedangkan *air to close* akan menutup apabila ada udara atau *supply* yang masuk pada *control valve*. *Control valve* terdiri atas 3 komponen penting yaitu: *actuator*, *body valve*, *positioner*. Gambar 2.1 adalah bagian aktuator *control valve*, sedangkan Gambar 2.2 yaitu bagian *body control valve*



**Gambar 2. 1** Aktuator *control valve*[3]



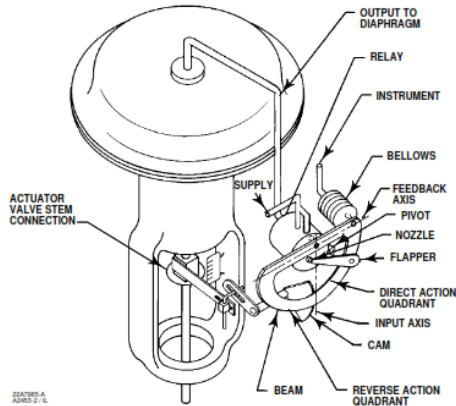
**Gambar 2. 2** *Body control valve*[3]

## 2.2 Positioner

*Positioner* adalah *controller* yang ada pada *control valve*, dimana nantinya sinyal keluaran dari *controller* akan diubah menjadi bukaan *valve*. *Positioner* memiliki PID kontrol tetapi sangat jarang dilakukan *tuning* pada *positioner*. Nilai yang biasa digunakan adalah P dengan nilai 100 dan I 20. 3 macam *positioner* yang digunakan pada *control valve* yaitu *pneumatic positioner*, *analog positioner* dan *digital controller* atau *smart positioner*.

- *Pneumatic positioner*: *Pneumatic positioner* menggunakan *input* sinyal *pneumatic* masuk pada *positioner*, yang nantinya akan dibuat untuk menggerakkan aktuator pada *control valve*.
- *Analog I/P positioner*: *Analog I/P positioner*, beda dengan *pneumatic positioner* adalah pada *supply* yang masuk pada *positioner*. *Analog I/P positioner* membutuhkan *input* arus listrik untuk dikonversi menjadi bukaan *valve*.
- *Smart positioner*: Jenis *positioner* ini sama dengan *analog I/P positioner* perbedaan ada pada konversi sinyal elektrik. Beberapa kelebihan *smart positioner* adalah: *supply* arus bisa dari *input* arus listrik 4-20 mA, otomatis kalibrasi

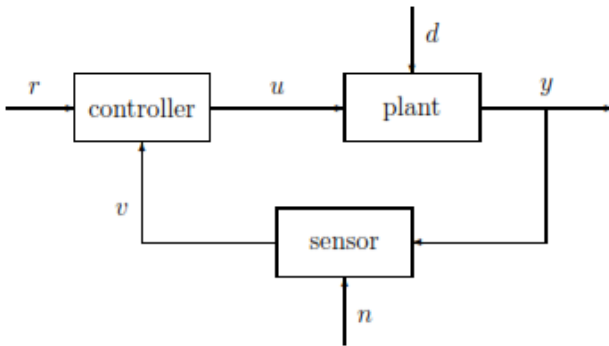
menggunakan tombol pada *positioner* atau HART, kemudahan dalam instalasi dan kalibrasi [3]



**Gambar 2. 3** *Positioner* [3]

### 2.3 Close Loop

*Closed loop* atau *feedback control* adalah sebuah system yang memiliki 3 komponen penting yaitu *plant*, *controller* dan sensor. *Plant* merupakan sebuah variabel atau alat yang nantinya akan dikontrol. Sensor atau dalam penelitian ini menggunakan *transmitter* digunakan untuk mengukur nilai keluaran sekaligus menjadi nilai masukan untuk *controller*. *Controller* adalah alat yang digunakan untuk menjaga proses *plant*. *Control valve* ada pada posisi antara *controller* dan *plant*. Keluaran *controller* menjadi masukan untuk *control valve*[4]. Keuntungan dalam sistem pengendalian *loop* tertutup adalah mempunyai respon yang cukup baik dengan konfigurasi yang *simple*.



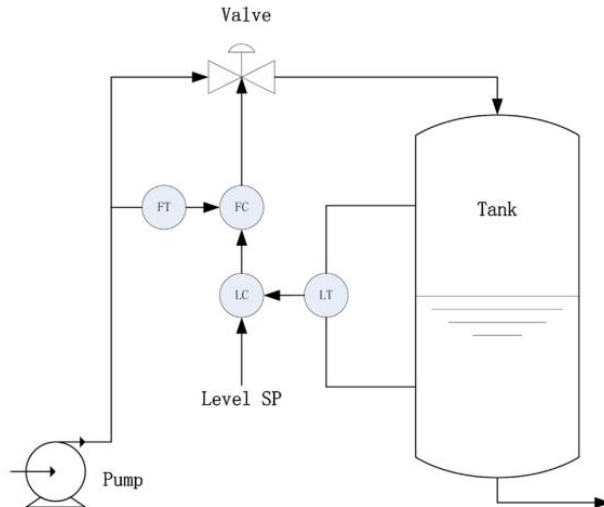
**Gambar 2. 4** Diagram *blok close loop* [5]

## 2.4 *Cascade control*

Sistem kontrol *cascade* merupakan sistem kontrol bertingkat yang digunakan untuk mengendalikan variabel keluaran *plant* yang saling mempengaruhi antara proses pertama dan proses kedua. *Cascade control* sendiri memiliki *master* dan *slave* kontrol dimana *slave* kontrol selalu responnya lebih cepat daripada *master*. Tujuan ditambahkan kontrol sekunder untuk mengurangi *disturbance* pada *output* sistem proses pertama yang dapat mempengaruhi proses selanjutnya, yaitu proses kedua. Fungsi dari kontrol *primer* untuk mengurangi *disturbance* pada *output* sistem proses kedua dan melemahkan sisa sinyal yang menuju proses kedua. Untuk pemilihan *secondary* dan *master* dilihat dari variabel yang mau dikontrol, variabel yang mempunyai respon lebih cepat akan dijadikan *secondary* dan yang lambat menjadi *master*. Contoh pengendalian *level* dan *flow* untuk *flow* menjadi *secondary* sedangkan *master* pada *level*. Penerapan sistem *cascade* dapat merugikan, apabila elemen proses di *primary loop* lebih cepat dari elemen proses pada *secondary loop*. Contoh apabila pada Gambar 2.5 *flow* akan kita gunakan untuk *master* dan *level* menjadi *slave*. Hal ini akan menimbulkan sistem yang cenderung beresilasi akibat interaksi antara *primary loop* dan *secondary loop*. Sehingga



sistem *cascade* hanya dapat diterapkan pada proses dengan *elemen primer* yang lebih lambat dari *elemen secondary* [6].



**Gambar 2. 5** Skema *cascade control* [7]

Gambar 2.5 menerangkan tentang skema atau P&ID untuk contoh pengendalian dengan menggunakan *Cascade control*. Skema tersebut menggunakan 2 *transmitter*, 2 *controller* dan 1 *control valve*.

Keuntungan pada sebuah *plant* yang menggunakan *cascade control* yakni:

- Respon keluaran dari *cascade control* lebih cepat daripada *single loop* kontrol
- Dengan adanya pengendali sekunder yang lebih cepat, dapat mengatasi pada gangguan sekunder.

Kerugian yang perlu dipertimbangkan dalam penggunaan *cascade control* adalah :

- Biaya atau rugi-rugi pengukuran variabel sekunder.
- Kerumitan pada konfigurasi apabila salah maka *error* yang terjadi lebih besar

- *Tuning* PID semakin susah dikarenakan mempunyai 2 variabel yang dikontrol

## 2.5 *Controller PID (Proportional Integral Derivative)*

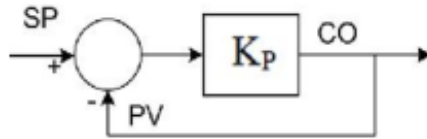
Sistem kontrol dikenal adanya sistem pengendalian PID merupakan suatu sistem pengendalian yang paling banyak digunakan didalam bidang industri. Sistem pengendalian PID telah diperkenalkan dari era sistem analog hingga menjadi sistem digital komputer. Perkembangan teknologi digital dan *software* mempengaruhi berkembangnya model kontrol PID[8].

Sistem kontrol PID merupakan sistem kontrol yang mengontrol dengan unsur *proporsional*, *integral*, dan *derivative*, yang berupa model matematika dengan masukan *Gain+Integral+Derivative*. PID merupakan sistem kontrol konvensional dalam bentuk matematis yang dapat merubah nilai *error* menjadi nol yaitu dengan menyamakan proses variabel sama dengan nilai *setpoint* sehingga kestabilan sistem kontrol dapat tercapai. Sistem kontrol PID merupakan sistem kontrol yang sangat unggul, namun dari ketiga parameter P,I, dan D masing-masing memiliki kekurangan dan kelebihan yang dapat mempengaruhi sinyal keluaran yang diinginkan[9].

Pengendalian *proportional* (P), merupakan aksi kontrol yang memiliki respon keluaran yang sebanding dengan sinyal kesalahan (*error*) yang dihasilkan, yang sesuai dengan persamaan dibawah ini.

$$CO(t) = K_p \cdot e(t) \quad (2.1)$$

Respon dari *proportional* dapat disesuaikan dengan nilai *error* terhadap  $K_p$  atau sesuai dengan *proportional gain*. Seperti pada Gambar 2.6 diagram blok *proportional*.

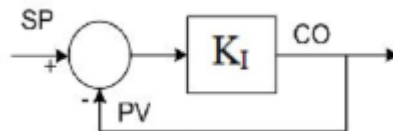


**Gambar 2. 6** Diagram blok pengendalian proporsional [8]

Pengendalian *Integral* (I) merupakan pengendalian yang bertujuan untuk menghilangkan *error steady state*. Persamaan pengendalian *integral* dapat dilihat pada persamaan dibawah ini.

$$CO(t) = K_i \int e(t) dt \quad (2.2)$$

*Integral* dapat menimbulkan respon yang berosilasi dengan amplitudo yang mengecil secara perlahan atau bahkan amplitudo akan membesar[8]. Diagram blok pengendalian *integral* dapat dilihat pada Gambar 2.7.



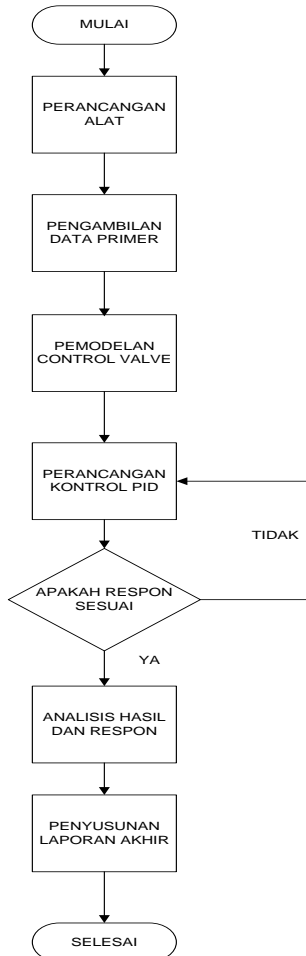
**Gambar 2. 7** Diagram blok pengendalian integral [8]

*Halaman ini sengaja dikosongkan*

## BAB III METODOLOGI PENELITIAN

### 3.1 Skema Diagram Alir Penelitian

Tahapan penelitian agar tujuan dari penelitian ini tercapai dilakukan digambarkan melalui *flowchart* yang ditunjukkan oleh Gambar 3.1.



**Gambar 3.1** *Flowchart* pengerjaan tugas akhir

**a. Perancangan *plant***

Perancangan yang dimaksud merupakan perancangan secara *hardware*. Perancangan tersebut dilakukan dengan membuat sebuah *plant* pengendalian *level* dengan *controller* menggunakan *national instrument, final control element* menggunakan *control valve* dan *DP transmitter*. Sensor *flow* digunakan sebagai pembacaan *flow rate* untuk pengambilan data.

**b. Pengambilan Data Primer**

Pengambilan data diambil secara langsung dari *plant* yang dibuat. Data yang diambil yaitu *pressure* yang masuk pada *positioner, open percent* dari *control valve, delta pressure control valve, flow outlet* dari *control valve*.

**c. Pemodelan *Control Valve***

Pemodelan *control valve* didapatkan dengan cara pengambilan data dari *pressure* yang masuk pada *positioner, open percent* dari *control valve, flow outlet* dari *control valve*.

**d. Perancangan kontrol PID**

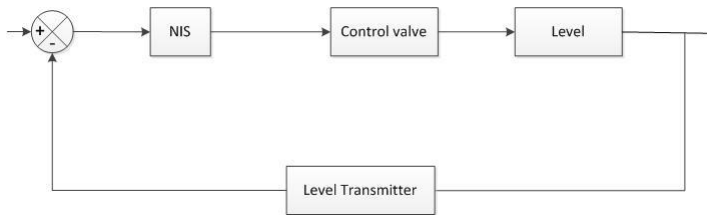
Perancangan kontrol akan menggunakan *auto tune* untuk menentukan nilai PID pada master dan PI pada *slave*.

**e. Analisis Hasil dan Respon**

Data hasil keseluruhan pengujian kemudian diolah serta dilakukan analisa setiap performansinya. Hasil pengolahan data tersebut kemudian dianalisa kelebihan dan kekurangannya.

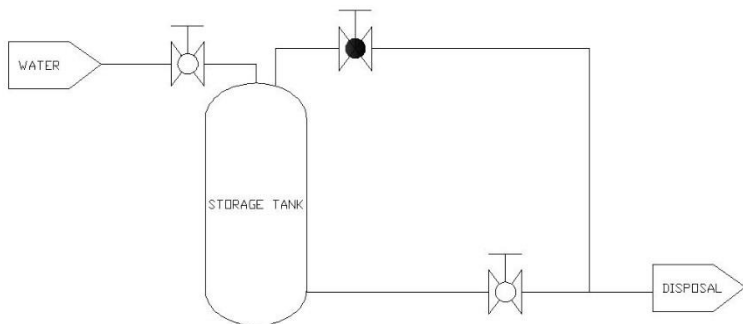
**3.2 Perancangan *plant***

Perancangan *plant* pengendalian *level* ini menggunakan *national instrument (NIS)* sebagai *controller*, sedangkan *control valve* digunakan sebagai *final control element (FCE)* dan *DP transmitter* sebagai *level transmitter*. *Flow* sensor digunakan sebagai pembacaan *flow rate* untuk pengambilan data. Diagram blok dapat dilihat pada gambar 3.2 dibawah ini.



**Gambar 3. 2** Diagram blok *close loop real plant*

*Plant* yang dirancang menggunakan masukan berupa air yang nantinya akan ditampung pada tangki. *Level* pada tangki akan dijaga dengan *controller* NIS dan *control valve* sebagai FCE. Proses *variable* akan dikirim oleh *level transmitter*. Gambar 3.3 adalah gambaran proses *flow diagram real plant*



**Gambar 3. 3** Proses *flow diagram plant*

*Plant level* pada penelitian ini menggunakan pipa berukuran 8 inch dan tinggi 1 meter untuk tangki. *Control valve* tipe *globe valve* dengan *smart positioner*. *Level transmitter* menggunakan *DP transmitter*. *Controller* menggunakan NIS dengan *software centum 3000* dan 3 *manual valve* digunakan nantinya untuk *distrubance*. Gambar 3.4 menunjukkan *real plant level*.



**Gambar 3. 4** *Plant level real plant*

Pemilihan ukuran *control valve* sangatlah penting pada suatu proses pengendalian, karena apabila *control valve* terlalu besar maka bukaan yang dihasilkan *control valve* terlalu kecil yang akan mengakibatkan *overshoot* yang tinggi dan mempercepat kerusakan pada *control valve*. Untuk memilih ukuran *control valve*, dapat menggunakan persamaan (3.1)

$$Q = cv \sqrt{\frac{\Delta p}{sg}} \quad (3.1)$$

Dengan:

$Q$  : *Flowrate* ( $m^3/s$ )

$Cv$ : *Koefisien Control Valve*

$\Delta P$  : *Perubahan Tekanan Control valve*(psi)

$sg$  : *Spesific Gravity*

Persamaan (3.1) di atas adalah dasar yang nantinya digunakan untuk memodelkan pemodelan *flow*. Pemodelan *flow* akan mewakili dari *body valve* dan pipa pada *real plant* yang nantinya dapat dicari untuk mencari nilai massa. *Flow* dikontrol *feedback* sudah dapat mewakili merubah *percentage control valve* menjadi massa sehingga mendapatkan respon yang lebih cepat.



Berdasarkan persamaan (3.1) dapat digunakan mencari nilai *mass flow rate* ( $m$ ) menggunakan persamaan sebagai berikut :

$$M = P.Q \quad (3.2)$$

Dengan:

$M$  : *Mass Flow Rate* (kg/s)

$P$  : *Massa Jenis Air* (kg/m<sup>3</sup>)

$Q$  : *Flow Rate* (m<sup>3</sup>/s)

### 3.3 Pemodelan Control Valve

*Control valve* digunakan sebagai penerima sinyal dari *controller* agar dapat mengatur berapa masukan yang mau dimanipulasi. Tahapan ini dilakukan pemodelan *control valve* melakukan pemodelan I/P, pemodelan aktuator, pemodelan *flow*, dan pemodelan *level*. Pemodelan dari *Control valve* didapatkan melalui pendekatan sistem orde satu, *First Order Plus Dead Time* (FOPDT), dengan menggunakan persamaan seperti dibawah ini [10]

$$G(s) = \frac{K e^{-\theta s}}{\tau s + 1} \quad (3.3)$$

Dengan:

$$K = \frac{\Delta y}{\Delta x}$$

$$\tau = 63\% \text{ rise time}$$

$K$  = *gain steady state*

$\Delta y$  = perubahan *steady state* pada keluaran proses

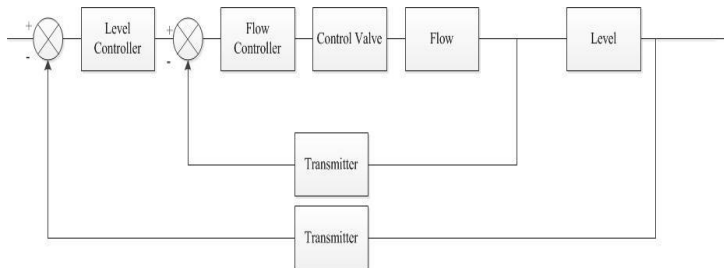
$\Delta x$  = perubahan *step* pada masukan

Parameter-parameter yang digunakan untuk pemodelan FOPDT adalah *process time constant* ( $\tau$ ) dan *process gain* ( $K$ ). Parameter-parameter diatas didapatkan melalui pengambilan data dengan memberikan *step response test* pada proses yaitu dengan memutus hubungan antara unit operasi dengan pengendali. Memutus hubungan antara proses dengan pengendali dapat dilakukan dengan mengubah mode *auto* pada pengendali menjadi mode *manual* atau yang biasa kita sebut *open loop*. Selanjutnya

memberikan perubahan pada variabel *input* dengan mengubah bukaan *valve*.

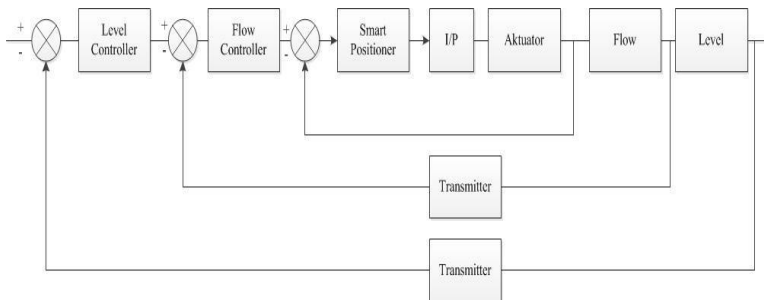
### 3.4 Perancangan *Control PID*

Tahapan ini dilakukan perancangan *control* pada *level* dan *PI Smart Positioner* untuk meningkatkan performansi dari pengendalian.



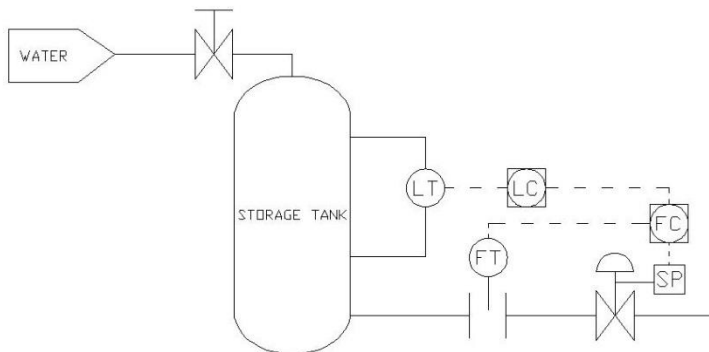
**Gambar 3. 5** Diagram Blok *Cascade control*

Gambar 3.5 di atas menerangkan tentang diagram blok *cascade* pada umumnya yaitu dengan 2 *transmitter*, 2 *controller* dan 1 aktuator. *Master* dan *slave* kontrol untuk dua pengendalian dimana kontrol *slave* tidak mungkin memiliki *controller* lebih bagus daripada *master* dan respon dari *slave* harus selalu lebih cepat daripada *master*.



**Gambar 3. 6** Diagram Blok *Cascade control Real plant*

Gambar 3.6 merupakan deskripsi dari *plant* yang digunakan pada penelitian ini yaitu *plant* dengan 2 *controller* dimana *level* sebagai *master* kontrol dan *flow* sebagai *slave* kontrol. Terdapat perbedaan pada Gambar 3.6 dan Gambar 3.5, dimana pada Gambar 3.6 dilakukan pemodelan *control valve* secara terpisah dimana aktuator dan *body valve* dimodelkan secara terpisah. Gambar 3.7 adalah P&ID dari *real plant*. Gambar di bawah menunjukkan rancangan *real plant* yang menggunakan *cascade control* dengan *level controller* sebagai *master* dan *flow controller* sebagai *slave*.

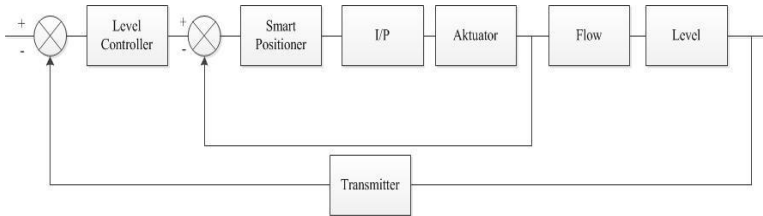


**Gambar 3. 7** P&ID *cascade control real plant*

Nilai KP dan KI pada *level controller* dan *smart positioner* didapatkan dengan melakukan *auto tune* pada *software Matlab*. Tabel 3.1 menunjukkan nilai KP dan KI pada *level controller* dan *smart positioner*

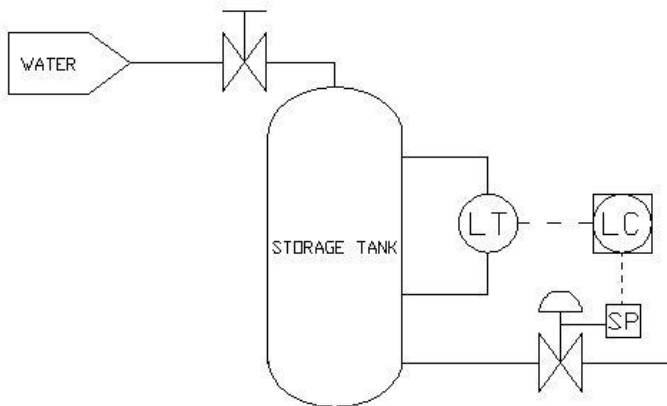
**Tabel 3. 1** Nilai KP dan KI pada *cascade control* dari *autotune Matlab*

	<i>Level controller</i>	<i>Flow controller</i>	<i>Smart positioner</i>
KP	0,072	14,817	0,211
KI	0,017	0,246	0,163



**Gambar 3. 8** Diagram blok *close loop level real plant*

Gambar 3.8 merupakan diagram blok *close loop level*, dengan tidak adanya *flow controller* maka respon yang didapatkan akan menjadi lebih lambat dikarenakan *disturbance* pada *flow* akan mempengaruhi *level controller*. Gambar 3.9 adalah P&ID dari *real plant* untuk *close loop level*, pada gambar tersebut dapat dilihat terjadi penghilangan pada *flow transmitter* dan *flow controller*.

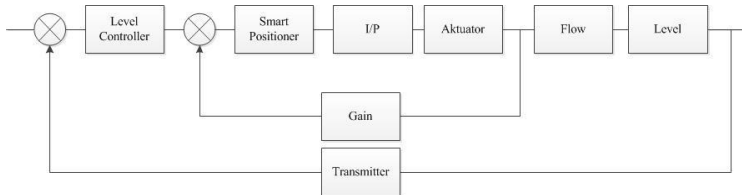


**Gambar 3. 9** P&ID *close loop level real plant*

Percobaan untuk mencari nilai KP dan KI pada metode *close loop level* didapatkan dengan melakukan metode yang sama dengan *cascade control* yaitu dengan *auto tune* pada *software matlab*. Tabel 3.2 menunjukkan nilai KP dan KI pada *level controller* dan *smart positioner*

**Tabel 3. 2** Nilai KP dan KI pada *close loop level* dari *autotune Matlab*

	<i>Level controller</i>	<i>Smart Positioner</i>
KP	2,535	0,083
KI	0,017	0,191

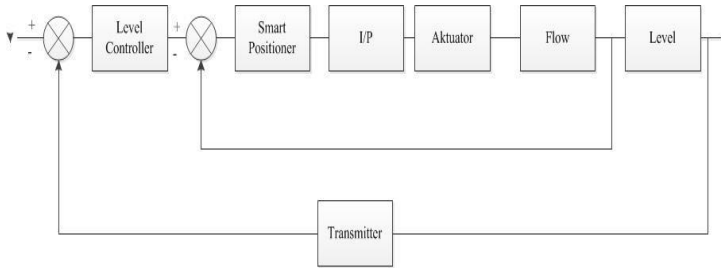


**Gambar 3. 10** Diagram blok penambahan *feedback gain flow*

Diagram blok pada Gambar 3.10 menunjukkan percobaan untuk membuktikan apabila diberi *feedback gain flow* pada *smart positioner* maka diagram blok di atas dapat mempunyai respon yang sama dengan *cascade control* pada gambar 3.6, dikarenakan *gain feedback* diatas mewakili nilai *flow*. Hasil respon pada Gambar 3.10 dapat dilihat pada Gambar 4.6. Tetapi tentunya ini tidak mungkin dapat diaplikasikan pada *real plant*. Tabel 3.3 mempunyai nilai yang sama dengan Tabel 3.2. Tabel 3.3 tidak dilakukan *tuning* ulang dikarenakan untuk membuktikan jika diberi *feedback gain flow* mempunyai respon yang sama dengan *cascade control*, *gain flow* bernilai 0,0284

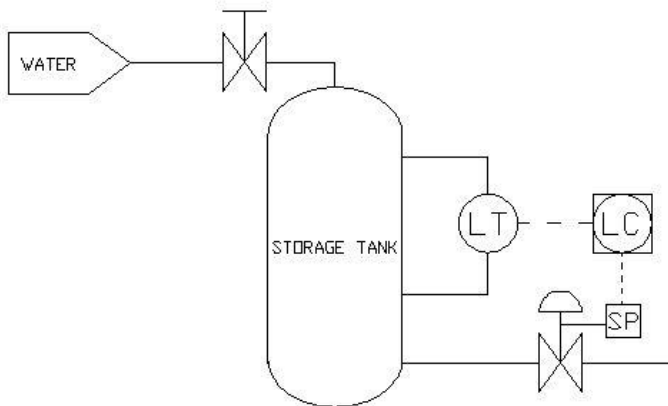
**Tabel 3. 3** Nilai KP dan KI pada *close loop level* dengan penambahan *gain flow* dari *autotune Matlab*

	<i>Level controller</i>	<i>Smart Positioner</i>
KP	2,535	0,083
KI	0,017	0,191



**Gambar 3. 11** Diagram blok cascade control sensorless

Gambar 3.11 adalah representasi *real plant* dari Gambar 3.10 atau metode *cascade control sensorless*. Nilai *gain feedback* pada Gambar 3.10 digunakan untuk menjadi pembagi dari nilai KP pada *Smart Positioner* dan pengali nilai KP untuk *level controller*. Dengan metode ini maka kita dapat mendapatkan respon yang sama dengan *cascade control*, dan dapat dilihat Gambar 3.12 mempunyai rancangan *real plant* yang sama dengan Gambar 3.9. Keuntungan yang didapatkan adalah dapat mengeliminasi 1 *transmitter* dan 1 *controller*



**Gambar 3. 12** P&ID cascade control sensorless level real plant

Nilai KP dan KI pada metode *cascade control sensorless* didapatkan dengan melakukan membagi nilai P dari *Smart Positioner* dengan *gain flow* dan mengalikan nilai P untuk *level controller*. Nilai KP dan KI yang digunakan untuk perhitungan gain adalah *close loop level*. Tabel 3.3 menunjukkan nilai KP dan KI pada *level controller* dan *smart positioner* yang telah dikalikan dan dibagi dengan *gain flow*

$$K_{P_{levelcontroller}} = 2,535 \times 0,0284 = 0,071$$

$$K_{P_{smartpositioner}} = 0,083 / 0,0284 = 2,922$$

**Tabel 3. 4** Nilai KP dan KI pada *cascade control sensorless* dari *autotune Matlab*

	<i>Level controller</i>	<i>Smart Positioner</i>
KP	0,071	2,922
KI	0,017	0,191

*Halaman Ini Sengaja Dikosongkan*



## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Perhitungan Control Valve

Data yang diperlukan untuk perhitungan *control valve sizing* yaitu *flow rate*, perbedaan tekanan, dan *specific gravity*. Hasil perbedaan tekanan diperoleh dari selisih antara *inlet* dan *outlet control valve*. *Specific gravity* memiliki nilai 1 dikarenakan fluida yang digunakan adalah air. Tabel 4.1 adalah hasil pengambilan data dan hasil *control valve sizing* perhitungan CV menggunakan persamaan (3.1). Persamaan (3.2) digunakan untuk mendapatkan data perhitungan *mass flow rate* yang dicantumkan pada Tabel 4.2.

**Tabel 4. 1** Hasil pengambilan data dan perhitungan pada *cv*

<i>Open Percentage (%)</i>	<i>Flow rate (m<sup>3</sup>/s)</i>	Perbedaan tekanan (psi)	<i>specific gravity</i>	CV
0	0	0	1	0
25	2,403	0,761	1	2,750
50	2,510	0,307	1	4,526
75	2,670	0,163	1	6,595
100	2,840	0,126	1	7,994

**Tabel 4. 2** Hasil perhitungan *mass flow rate*

<i>Mass flow rate (kg/s)</i>	<i>Flow rate (m<sup>3</sup>/s)</i>	Massa jenis (kg/m <sup>3</sup> )
0	0	1
2,403	2,403	1
2,510	2,510	1
2,670	2,670	1
2,840	2,840	1

## 4.2 Pemodelan Plant dan Control Valve

Pengambilan data primer diambil langsung dari *plant* pengendalian *level*, *flow*, dan setiap komponen pada *control valve*. Data yang didapatkan dari *plant* digunakan untuk membuat pemodelan setiap komponen pada setiap diagram blok. Pengambilan data *time constant* digunakan untuk pemodelan setiap komponen. Hasil pengambilan data *time constant* dapat dilihat pada Tabel 4.3.

**Tabel 4. 3** Hasil pengambilan data *time constant*

Komponen	<i>Time Constant (second)</i>
I/P	1,449
Aktuator	12,0771
<i>Flow</i>	0,5292

### a. Pemodelan I/P

I/P memiliki fungsi untuk mengubah sinyal listrik 4 mA – 20 mA dari *controller* menjadi sinyal *pneumatik* 3 psi – 15 psi. Pengambilan data pada I/P dilakukan pada setiap variasi arus yang dilakukan dan pada *pressure* yang masuk pada *smart positioner*. Hasil pengambilan data untuk I/P ditunjukkan pada Tabel 4.4.

**Tabel 4. 4** Hasil pengambilan data pada I/P

Arus yang diinputkan (ma)	Arus pada <i>smart positioner</i> (ma)	<i>Pressure</i> pada <i>smart positioner</i> (psi)
4	4	0
8	7,98	11,50575
12	12	17,139
16	16,02	22, 8085
20	20,03	28,0517

Hasil pengambilan data I/P menunjukkan nilai *error* yang ada pada *control valve* tidak terlalu besar, sehingga dapat dinilai bahwa *smart positioner* masih dinilai baik. Nilai pemodelan yang didapatkan adalah:

$$\text{FOPDT}(i/p) = \frac{(28,0517-0)/(20-16)}{1,449s+1} = \frac{1,74995}{1,449s+1} \quad (4.1)$$

### b. Pemodelan Aktuator

Pengambilan data pada aktuator diambil pada *pressure* yang masuk pada *control valve* dan berapa persen bukaan dari *control valve*. Hasil pengambilan data untuk aktuator ditunjukkan pada Tabel 4.5.

**Tabel 4. 5** Hasil pengambilan data pada aktuator

<i>Pressure</i> pada <i>smart positioner</i> (psi)	<i>Open percentage</i> pada <i>control valve</i> (%)
0	0
11,50575	24,9
17,139	49,7
22, 8085	75,3
28,0517	100

Pengambilan data diambil dari arus 4-20 mA, untuk *pressure* yang didapatkan dilihat dari *smart positioner* pada *control valve*. Pemodelan yang didapatkan sebagai berikut :

$$\text{FOPDT}(\text{aktuator}) = \frac{(100-0)/(28,0517-0)}{12,0771s+1} = \frac{3,347}{12,0771s+1} \quad (4.2)$$

### c. Pemodelan *flow*

Pengambilan data pada *flow* dilakukan pada setiap variasi bukaan *valve*. Hasil pengambilan data untuk *flow* ditunjukkan pada Tabel 4.6.

**Tabel 4. 6** Hasil pengambilan data pada flow

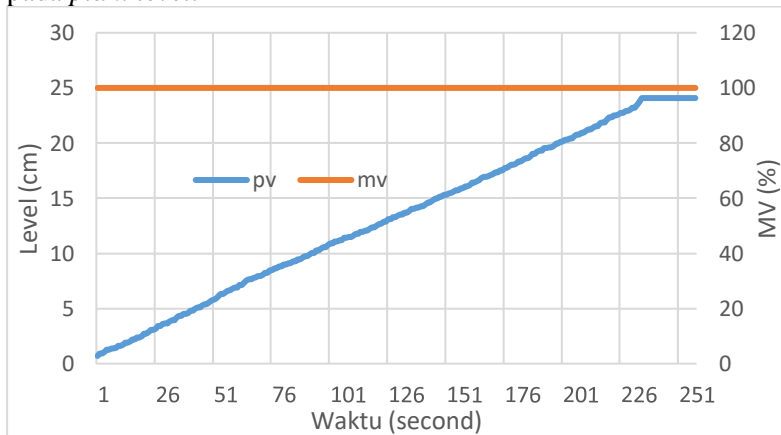
Open percentage pada control valve (%)	Flow rate (m <sup>3</sup> /s)
0	0
24,9	2,4
49,7	2,51
75,3	2,67
100	2,84

Pemodelan *flow* ini sebenarnya adalah pemodelan dari *body valve* dan pipa pada *plant*. Pemodelan yang dimaksud adalah sebagai berikut:

$$\text{FOPDT}(\text{flow}) = \frac{(2,84-0)/(100-0)}{0,5292s+1} = \frac{0,0284}{0,5292s+1} \quad (4.3)$$

#### d. Pemodelan level

Pemodelan pada *level* dilakukan dengan cara melakukan uji *open loop* pada *real plant*, untuk mendapatkan nilai *gain* dan *time constant*. Gambar 4.1 adalah respon uji *open loop* yang dilakukan pada *plant level*.

**Gambar 4.1** Uji *open loop level*

Berikut ini pemodelan yang didapat untuk plant *level* adalah sebagai berikut:

$$\text{FOPDT}(\text{level}) = \frac{(23,66-0)/(2,84-0)}{(0,63 \times 236)s+1} = \frac{8,333}{148,68s+1} \quad (4.4)$$

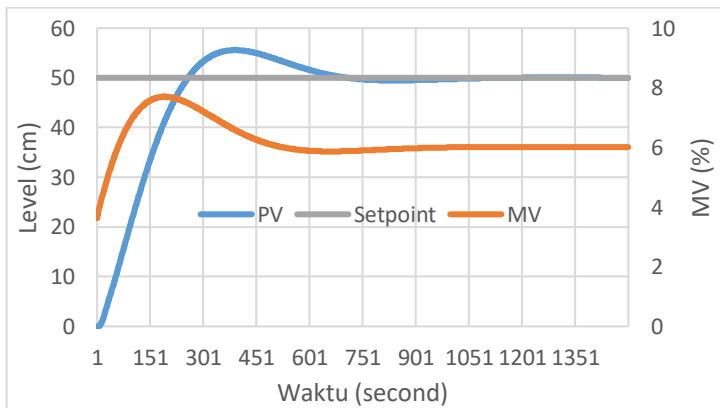
#### e. Pemodelan *level transmitter*

Pemodelan pada *level transmitter* bernilai 1, dikarenakan *delay time* dan *time constant* bernilai 0. Hal ini disebabkan karena kerja *level transmitter* yang cepat, sehingga pemodelan untuk *level transmitter* pada simulink dapat diabaikan. Hasil pemodelan *level transmitter* sebagai berikut:

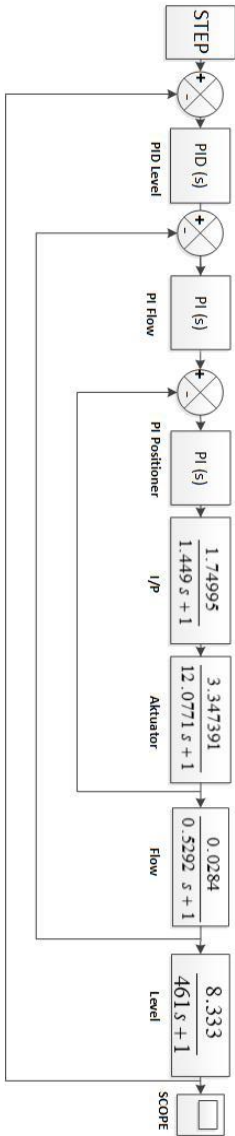
$$\text{FOPDT}(\text{level transmitter}) = \frac{1}{(0,63 \times 0)s+1} = 1 \quad (4.5)$$

### 4.2 Hasil Simulink *Cascade Control*

Diagram blok *cascade control* menggunakan 2 *transmitter*, 2 *controller* dan 1 aktuatur. *Control valve* sendiri dipecah menjadi 3 blok yaitu: *positioner*, I/P aktuatur. *Controller* berupa *positioner* dan I/P dan aktuatur sebagai penggerak pada *control valve*. Gambar 4.2 adalah hasil grafik respon simulasi *software* Simulink. Hasil Pemodelan *cascade control* pada *real plant* berdasarkan persamaan pemodelan direpresentasikan dalam *software Matlab* pada Gambar 4.3.



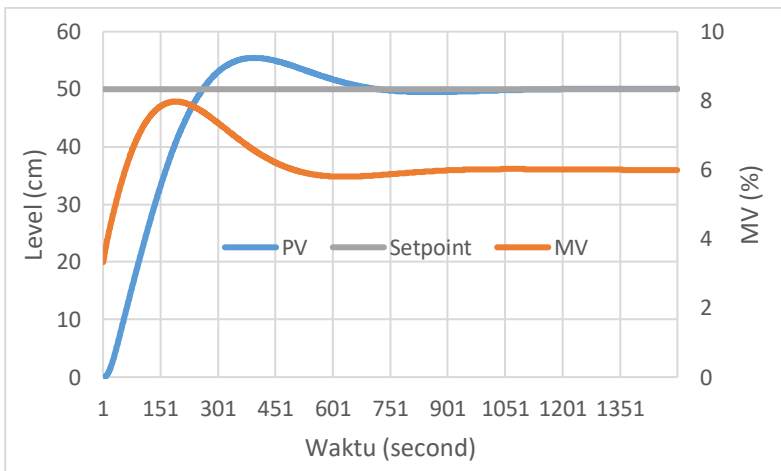
**Gambar 4. 2** Hasil respon *simulink cascade control*



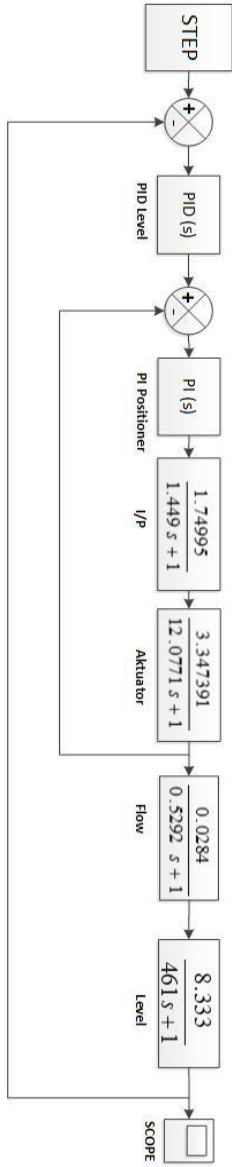
**Gambar 4. 3** *Pemodelan cascade control real plant*

### 4.3 Hasil Simulink Close Loop Level

*Close loop level* pada simulink masih terdapat kontrol PI pada *smart positioner*, tetapi *feedback* kontrol tidak pada *flow* melainkan pada aktuatur *control valve* sehingga masih dapat dibilang sebagai *close loop*. Gambar 4.4 adalah hasil dari simulasi pada Gambar 4.5 yang menghasilkan grafik respon *close loop level*. Hasil Pemodelan *close loop level* direpresentasikan dalam software *Matlab* pada Gambar 4.5 seperti berikut:



**Gambar 4. 4** Hasil respon *simulink close loop level control*

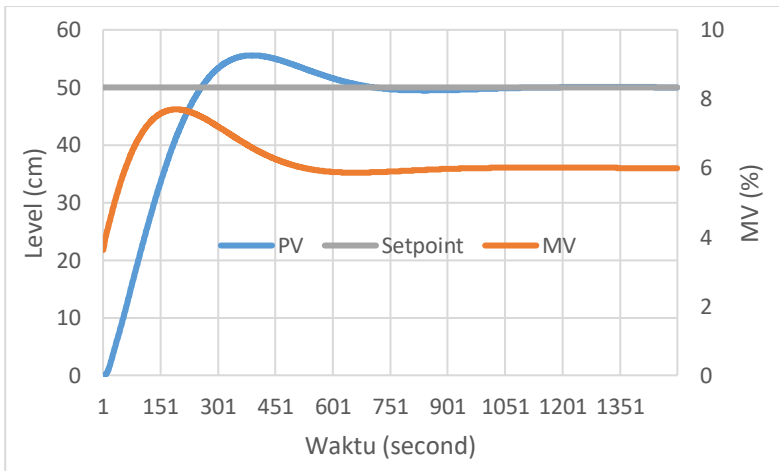


**Gambar 4. 5** *Penodelan close loop level*

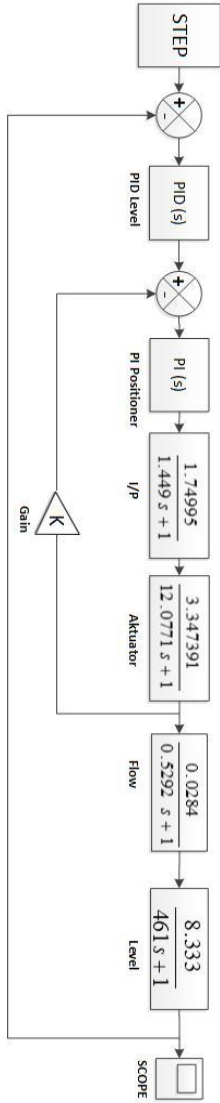


#### 4.4 Hasil Simulink *Close Loop Level* Dengan Konstanta

Pemodelan *Close Loop Level* dengan menggunakan konstanta bertujuan untuk membuktikan bahwa dengan menggunakan nilai *gain flow* sebagai *feedback* dapat memberikan respon *close loop level* menjadi sama dengan *cascade control*. Gambar 4.6 menunjukkan grafik simulasi *close loop level* dengan konstanta, dimana konstanta bernilai *gain flow*. Hasil respon grafik dapat dilihat sama dengan *cascade control*, maka dapat disimpulkan bahwa *PI positioner* bisa menjadi *feedback control* pengganti *secondary control flow* pada *cascade control*. Gambar 4.7 adalah representasi hasil simulasi menggunakan *software Matlab* sebagai berikut :



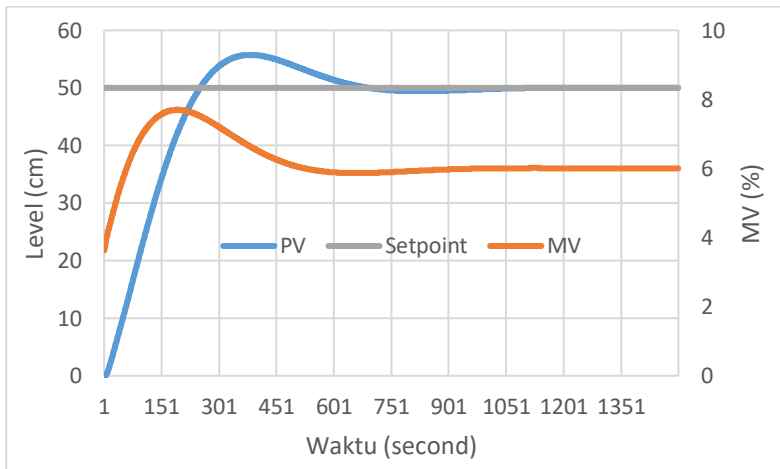
**Gambar 4. 6** Hasil respon *simulink close loop level* dengan konstanta



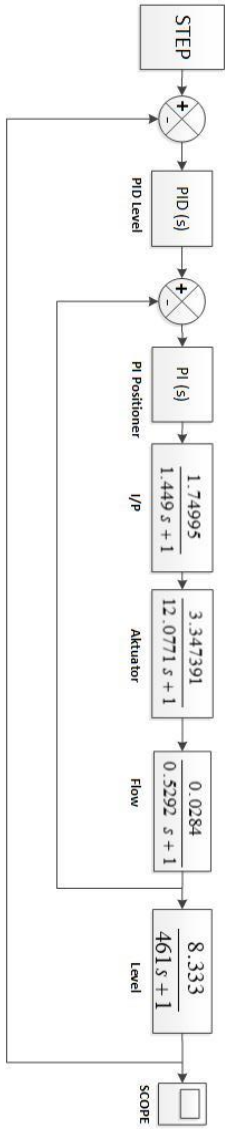
**Gambar 4. 7** *Pemodelan close loop level dengan konstanta*

#### 4.5 Hasil Pemodelan Cascade Control Sensorless

*Cascade control sensorless* bertujuan untuk menghilangkan *flow control* dan *flow transmitter* tetapi tidak memperlambat respon bahkan mempunyai respon yang sama dengan *cascade control*, dikarenakan nilai *gain flow* pada Gambar 4.7 digunakan sebagai pembagi dan pengali nilai konstanta P dari *level control* dan *smart potitioner*. Hasil simulasi menghasilkan grafik respon dari pemodelan *cascade control sensorless* yang direpresentasikan pada Gambar 4.8. Grafik respon menunjukkan *cascade control sensorless* menunjukkan hasil yang sama dengan *cascade control*, sehingga *cascade control sensorless* dapat diaplikasikan sebagai pengganti *cascade control*. Berikut hasil pemodelan *cascade control sensorless* dalam software Matlab pada Gambar 4.9.



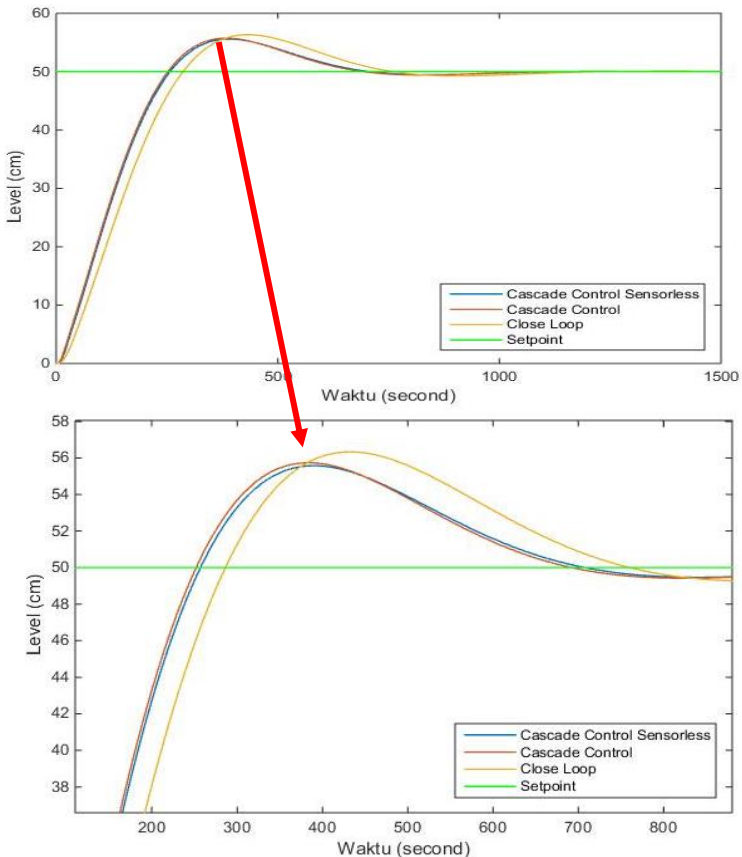
**Gambar 4. 8** Hasil respon *simulink cascade control sensorless*



**Gambar 4.9** *Pemodelan cascade control sensorless*

#### 4.6 Perbandingan Hasil Respon Simulink

Hasil dari semua simulasi yang telah dilakukan kemudian dilakukan analisa respon kualitatif dan kuantitatif, sehingga dapat menyimpulkan bahwa *cascade control sensorless* mempunyai respon yang hampir sama dengan *cascade control*. Gambar 4.10 adalah perbandingan dari setiap metode yang telah dilakukan mulai dari *close loop*, *cascade control* dan *cascade control sensorless*.



**Gambar 4. 10** Perbandingan hasil respon *simulink close loop level, cascade control , dan cascade control sensorless*

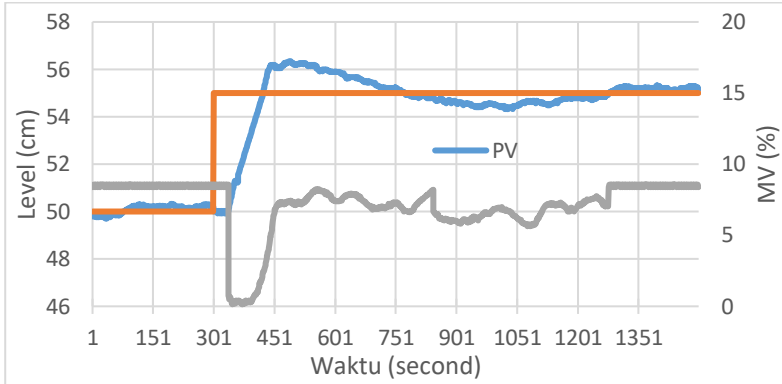
Tabel 4.7 menunjukkan nilai dari analisis respon kualitatif dan kuantitatif pada Gambar 4.10, didapatkan hasil *tuning* didapatkan data nilai respon dari *close loop level*, *cascade control*, *close loop level* dengan *gain*, dan *cascade control sensorless*. Hasil *tuning close loop level* didapatkan nilai *maximum overshoot* 6,009 cm dengan nilai 11,743 %, dan *settling time* 2190,7 *second*. Untuk hasil *tuning cascade control* didapatkan nilai *maximum overshoot* 5,770 cm dengan nilai 11,540 %, dan *settling time* 1931,5 *second*. Hasil *tuning cascade control sensorless* didapatkan nilai *maximum overshoot* 5,734 cm dengan nilai 11,469 %, dan *settling time* 1981,8 *second*.

**Tabel 4. 7** Perbandingan nilai grafik

	Max Overshoot (cm)	Max Overshoot (%)	Settling Time (s)	Error Ready State (%)	IAE
<i>Close Loop</i>	6,009	11,743	2190	5	247
<i>Cascade Control</i>	5,770	11,540	1931	5	223
<i>Cascade Control Sensorless</i>	5,734	11,469	1981	5	217

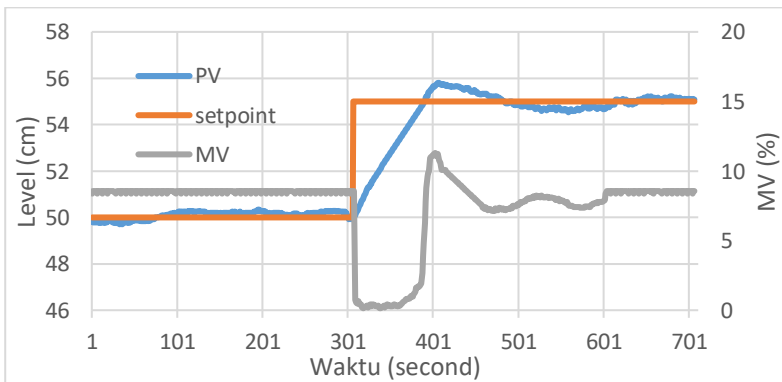
#### 4.7 Perbandingan Hasil Respon *Real Plant*

Perbandingan Hasil respon *real plant* bertujuan untuk membandingkan antara respon *close loop* dan *cascade control sensorless* secara kualitatif dan kuantitatif dengan cara uji *setpoint* dan uji *disturbance*. Gambar 4.11 adalah grafik uji *setpoint close loop level* yang diambil dari *real plant*. Pengambilan data *level* dimulai dari ketinggian air 50 cm dengan *setpoint* 50 cm dan 55 cm.



**Gambar 4. 11** Respon *close loop level* saat diubah *setpoint* dari 50 cm dan 55 cm

Grafik respon uji *setpoint cascade control sensorless* ditunjukkan pada Gambar 4.12 dengan pengambilan data yang dilakukan secara *real plant*. Pengambilan data dimulai dari ketinggian air 50 cm dengan *setpoint* 50 cm dan 55 cm.



**Gambar 4. 12** Respon *cascade control sensorless* saat diubah *setpoint* dari 50 cm dan 55 cm

Berdasarkan grafik yang ditunjukkan pada Gambar 4.11 dan 4.12 kemudian dilakukan analisis respon yang dapat dilihat pada Tabel 4.8 sebagai berikut:

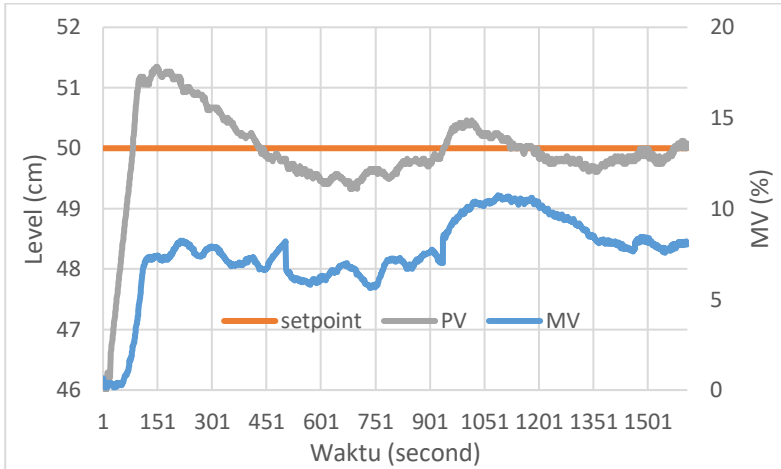
**Tabel 4. 8** Perbandingan nilai grafik real plant

<i>Set point 55 cm</i>					
	<i>Max Overshoot (m)</i>	<i>Max Overshoot (%)</i>	<i>Settling Time (s)</i>	<i>Error Steady State (%)</i>	<i>IAE</i>
<i>Close Loop</i>	1,355	2,460	560	2	9,383
<i>Cascade Control Sensorless</i>	0,80	1,454	408	1,454	3,704

Tabel 4.8 dapat didapatkan hasil bahwa pengendalian level menggunakan *Cascade Control Sensorless* memiliki respon yang lebih cepat dengan nilai *settling time* masing masing *set point* sebesar 408 *second* dan 560 *second* dibandingkan dengan pengendalian level secara *close loop*.

Gambar 4.13 merupakan grafik uji *disturbance close loop* yang diambil secara *real plant*. Data diambil dari ketinggian air 45 cm dengan *setpoint* 50 cm, ketika sudah stabil pada 50 cm akan diberikan *disturbance* dengan memberikan *input* 2 kali lebih besar dari *flow* normal selama 30 detik.

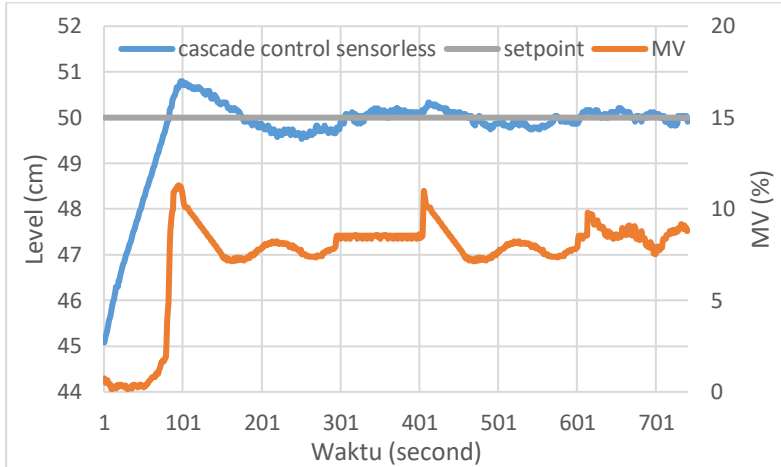




**Gambar 4. 13** Hasil respon *close loop level* dengan diberi *disturbance*

Gambar 4.13 menunjukkan bahwa uji *disturbance* diberikan pada detik 980 selama 30 detik sehingga dilihat dari respon *close loop level* meskipun terdapat *disturbance* respon tetap kembali pada nilai *steady*.

Gambar 4.14 menunjukkan grafik uji *disturbance cascade control sensorless level* yang diambil dari *real plant*. Data diambil dari ketinggian 45 cm dengan *setpoint* 50 cm, ketika sudah stabil pada 50 cm akan diberikan *disturbance* dengan memberikan *input* 2 kali lebih besar dari *flow* normal selama 30 detik .



**Gambar 4. 14** Hasil respon *cascade control sensorless* dengan diberi *disturbance*

Gambar 4.14 dapat dilihat tidak terdapat perubahan yang signifikan apabila terdapat *disturbance* pada detik 420 selama 30 detik yang diberikan pada *plant* karena respon tetap kembali ke nilai *steady*.

Berdasarkan grafik yang ditunjukkan pada Gambar 4.13 dan 4.14 kemudian dilakukan analisis respon yang dapat dilihat pada Tabel 4.9 sebagai berikut:

**Tabel 4. 9** Perbandingan nilai grafik real plant

	<i>Set point 50 cm</i>				
	<i>Max Overshoot (cm)</i>	<i>Max Overshoot (%)</i>	<i>Settling Time (s)</i>	<i>Error Steady State (%)</i>	<i>IAE</i>
<i>Close Loop</i>	1,385	2,463	230	2	9,567
<i>Cascade Control Sensorless</i>	0,818	1,459	99	1,454	3,904

## **BAB V PENUTUP**

### **5.1 Kesimpulan**

Berdasarkan hasil penelitian yang dilakukan pada tugas akhir ini, didapatkan kesimpulan sebagai berikut ini:

- a. *Cascade control sensorless* yang dirancang dapat menggantikan *control flow* dengan memanfaatkan *gain flow* sebesar 0,0284 sebagai pembagi nilai KP dari *Smart Positioner* dan pengali nilai KP untuk *level controller*
- b. Hasil kinerja sistem ditunjukkan melalui perbandingan *tuning cascade control* dan *cascade control sensorless*. Hasil yang didapatkan *cascade control* yaitu nilai *maximum overshoot* 5,770 cm dengan nilai 11,540 %, dan *settling time* 1931,5 sekon. Hasil *tuning cascade control sensorless* didapatkan nilai *maximum overshoot* 5,734 cm dengan nilai 11,469 %, dan *settling time* 1981,8 sekon.

### **5.2 Saran**

Saran dari penelitian kali ini yaitu penelitian ini seharusnya bisa dikembangkan di *plant* yang besar.

*Halaman Ini Sengaja Dikosongkan*

## DAFTAR PUSTAKA

- [1] Miao T and Seborg D.E., "Automatic detection of excessively oscillatory feedback control loops," *Proceedings of the 1999 IEEE International Conference on Control Applications, 1999, vol.1*, pp. 359, 1999.
- [2] M. Shamsuzzoha and S. Skogestad, "The *setpoint* overshoot method: A simple and fast closed loop approach for PID tuning," *Journal of Process Control*, p. 1220, 2010.
- [3] LLC Fisher Controls International, *Control Valve Handbook Fourth Edition*, Fisher, 2005.
- [4] Pertamina, *Dasar Instrumentasi dan Proses Kontrol*, 2007.
- [5] J. Doyle dan A. T. Bruce Francis, *Feedback Control Theory*, Macmillan, 1990.
- [6] P. Ahusda and K. Indriawati, "Penerapan Kontrol Optimal LQG Pada Sistem Kontrol Cascade PI&P Untuk Pengendalian Temperature di Superheater".
- [7] Chen Li, S. Choudhury, B. Huang dan F. Qian, "Frequency analysis and compensation of valve stiction in cascade control loops," *Journal of Process Control*, 2014.
- [8] Bhakti Yudho. S, H. Hikmarika, S. Dwijayanti and Purwanto, "Aplikasi Perbandingan Pengendali P, PI, Dan PID Pada Proses Pengendalian Suhu Dalam Sistem Mini Boiler," *Jurnal amplifier*, vol. 3, 2013.
- [9] R. Maerani and S. Bahkri, "Perbandingan Sistem Pengontrolan Pid Konvensional Dengan Pengontrolan Cmac, Fuzzy Logic Dan Ann Pada Water Level," vol. 17, pp. 129-141, 2013.
- [10] R. Rivaldi, *Tuning Kontrol PID Pada Ole Process Control (OPC) Server Dan Distributed Control System*, Surabaya : ITS, 2016.



## **BIODATA PENULIS**



Penulis dilahirkan di Surabaya pada tanggal 09 Maret 1994. Penulis berasal dari Kedung Rukem 4 no 54. Pada tahun 2006, penulis telah menyelesaikan pendidikan di SDN Kaliasin 1. Pada tahun 2009, penulis telah menyelesaikan pendidikan tingkat menengah pertama di SMPN 1 Surabaya. Tahun 2012 penulis telah menyelesaikan pendidikan di SMAN 2 Surabaya. Tahun 2015, penulis mampu menyelesaikan pendidikan Diploma di Program Studi D3

Metrologi dan Instrumentasi, Jurusan Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya. Tahun 2017, penulis melanjutkan pendidikan sarjana di program lintas jalur Teknik Fisika, Jurusan Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya. Penulis dapat dihubungi melalui email [muhamadmuiz15@gmail.com](mailto:muhamadmuiz15@gmail.com).