Logo

Description automatically generated

**TUGAS AKHIR –** **EE184801**

**PERANCANGAN KONTROLER PID DENGAN METODE *PARTICLE SWARM OPTIMIZATION* UNTUK SISTEM PENGATURAN CASCADE LEVEL DAN FLOW PADA PLANT PCT-100**

**KHOIRUS FAUZI RAHMADHANI**

NRP 07111840000029

Dosen Pembimbing

**Eka Iskandar, ST., MT**

NIP 198005282008121001

**Program Studi Teknik Sistem Pengaturan**

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2022

Icon

Description automatically generated

**TUGAS AKHIR – EE184801**

**PERANCANGAN KONTROLER PID DENGAN METODE PARTICLE SWARM OPTIMIZATION UNTUK SISTEM PENGATURAN CASCADE LEVEL DAN FLOW PADA PLANT PCT-100**

**KHOIRUS FAUZI RAHMADHANI**

NRP 07111840000029

Dosen Pembimbing

**Eka Iskandar, ST., MT**

NIP 198005282008121001

**Program Studi Teknik Sistem Pengaturan**

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2022

Icon

Description automatically generated

**FINAL PROJECT – EE184801**

**PID CONTROLLER DESIGN WITH PARTICLE SWARM OPTIMIZATION METHOD FOR CASCADE LEVEL AND FLOW CONTROL SYSTEMS AT PLANT PCT-100**

**KHOIRUS FAUZI RAHMADHANI**

NRP 07111840000029

Advisor

**Eka Iskandar, ST., MT**

NIP 198005282008121001

**Study Program Control System Engineering**

Department of Electrical Engineering

Faculty of Intelligent Electrical and Information Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2022

LEMBAR PENGESAHAN

**PERANCANGAN KONTROLER PID DI DCS YOKOGAWA CENTUM VP UNTUK SISTEM PENGATURAN CASCADE LEVEL DAN FLOW PADA PLANT PCT-100**

**TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat

memperoleh gelar Sarjana Teknik pada

Program Studi S-1 Teknik Sistem Pengaturan

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Oleh: KHOIRUS FAUZI RAHMADHANI

NRP. 07111840000029

Disetujui oleh Tim Penguji Tugas Akhir:

|  |  |  |
| --- | --- | --- |
| 1. | Eka Iskandar, ST., MT | Pembimbing |
| 2. | Mohamad Abdul Hady,S.T.,M.T. | Ko-pembimbing |
| 3. | Nama dan gelar penguji | Penguji |
| 4. | Nama dan gelar penguji | Penguji |
| 5. | Nama dan gelar penguji | Penguji |

**SURABAYA**

**Desember, 2022**

Text, letter

Description automatically generated

ABSTRAK

**ANALISA PENGARUH PANJANG LINKAGE TERHADAP RESPON SERIES ACTIVE VARIABLE GEOMETRY SUSPENSION (SAVGS)**

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa / NRP** | **:** | **Khoirus Fauzi Rahmadhani / 07111840000029** |
| **Departemen** | **:** | **Teknik Elektro FTEIC - ITS** |
| **Dosen Pembimbing** | **:** | **Eka Iskandar, ST., MT** |

**Abstrak**

Dalam sistem kontrol proses industri skala besar, variabel yang sering dikontrol adalah level cairan, laju aliran, suhu, tekanan dan sebagainya, yang sebagian besar merupakan fungsi non linier dari input kontrol. Namun, untuk menggunakan teknik kontrol linear secara umum proses nonlinier perlu didekati untuk mendapatkan model linierisasi. Lebih dari 90% aplikasi kontrol industri didasarkan pada mekanisme kontrol PID, seperti sistem robotika dan pengendalian otomatis. Namun dalam system pengaturan yang terdapat controller PID masih belum optimal terutama adanya disturbance atau gangguan berupa ketidakstabilan laju aliran pada variable yang dimanipulasi sehingga output juga berubah. Kelemahan dari pengontrol konvensional tidak dapat diabaikan karena sebagian besar industri berfokus pada akurasi, margin keuntungan, dan waktu operasi. Tujuan dari tugas akhir ini adalah merancang algoritma tuning PID dengan metode particle swarm optimization. Dari tugas akhir ini diperoleh hasil tuning yang optimal dan dan dapat memperkecil error steady state sebesar 80%.

**Kata kunci: *Level, Flow, PID, Particle Swarm Optimization****.*

ABSTRACT

**ANALYSIS OF THE EFFECT OF LINKAGE LENGTH ON SERIES ACTIVE VARIABLE GEOMETRY SUSPENSION (SAVGS) RESPONSE**

|  |  |  |
| --- | --- | --- |
| **Student Name / NRP** | **:** | **Khoirus Fauzi Rahmadhani / 07111840000029** |
| **Department** | **:** | **Teknik Elektro FTEIC - ITS** |
| **Advisor** | **:** | **Eka Iskandar, ST., MT** |

**Abstract**

**Keywords: *LQR, Quarter-car, SAVGS, Simscape Multibody, Suspension****.*

DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc121988121)

[PERNYATAAN ORISINALITAS ii](#_Toc121988122)

[ABSTRAK iii](#_Toc121988123)

[ABSTRACT iv](#_Toc121988124)

[DAFTAR ISI v](#_Toc121988125)

[DAFTAR GAMBAR vi](#_Toc121988126)

[DAFTAR TABEL vii](#_Toc121988127)

[DAFTAR SIMBOL viii](#_Toc121988128)

[BAB 1 PENDAHULUAN 1](#_Toc121988129)

[1.1 Latar Belakang 1](#_Toc121988130)

[1.2 Rumusan Masalah 1](#_Toc121988131)

[1.3 Batasan Masalah 2](#_Toc121988132)

[1.4 Tujuan 2](#_Toc121988133)

[1.5 Manfaat 2](#_Toc121988134)

[BAB 2 TINJAUAN PUSTAKA 4](#_Toc121988135)

[2.1 Hasil Penelitian Terdahulu 4](#_Toc121988136)

[2.2 Plant PCT 100 4](#_Toc121988137)

[2.3 Pemodelan *Plant* PCT-100 5](#_Toc121988138)

[2.3.1 Pemodelan Pompa 7](#_Toc121988139)

[2.3.2 Pemodelan Pipa 7](#_Toc121988140)

[2.3.3 Pemodelan Tanki 7](#_Toc121988141)

[2.3.4 Pemodelan Sensor 9](#_Toc121988142)

[2.4 Pemodelan Sistem Dengan Menggunakan Kurva Reaksi 9](#_Toc121988143)

[2.4.1 Menentukan jumlah dari parameter dan 10](#_Toc121988144)

[2.4.2 Menentukan parameter dan 10](#_Toc121988145)

[2.4.3 Menentukan parameter pada kurva Harriot 10](#_Toc121988146)

[2.4.4 Menentukan parameter 10](#_Toc121988147)

[2.4.5 Menentukan Parameter 11](#_Toc121988148)

[2.4.6 Menentukan parameter 11](#_Toc121988149)

[2.5 Kontroler PID 11](#_Toc121988150)

[2.6 Sistem Pengaturan *Cascade* 12](#_Toc121988151)

[*2.7* *Particle Swarm Optimization* 12](#_Toc121988152)

[BAB 3 METODOLOGI 15](#_Toc121988153)

[3.1 Metode yang digunakan 15](#_Toc121988154)

[3.2 Bahan dan peralatan yang digunakan 16](#_Toc121988155)

[3.2.1 Process Control Technology 100 Byotronics 16](#_Toc121988156)

[3.2.2 Adam 5000 L TCP/IP 16](#_Toc121988157)

[3.2.3 Matlab 17](#_Toc121988158)

[3.2.4 Simulink 17](#_Toc121988159)

[3.2.5 Personal Computer 17](#_Toc121988160)

[3.3 Urutan pelaksanaan penelitian 18](#_Toc121988161)

[3.3.1 Pemodelan Sistem Pengaturan Level pada PCT 100 18](#_Toc121988162)

[3.3.2 Pemodelan Sistem Pengaturan Flow 19](#_Toc121988163)

[3.3.3 Tuning PID Menggunakan pada Sistem Pengaturan Cascade 23](#_Toc121988164)

[3.3.4 Tuning PID dengan Metode Particle Swarm Optimization 24](#_Toc121988165)

[BAB 4 Hasil Penelitian dan Pembahasan 27](#_Toc121988166)

[4.1 Hasil Pemodelan dari Sistem Pengaturan Level 27](#_Toc121988167)

[4.2 Hasil Pemodelan dari Sistem dari Sistem Pengaturan Flow 27](#_Toc121988168)

[4.3 Hasil Pengujian Tuning PID dengan PSO pada Sistem Pengaturan Level 29](#_Toc121988169)

[4.4 Hasil Pengujian Tuning PID dengan PSO pada Sistem Pengaturan Flow 29](#_Toc121988170)

[4.4.1 Pengujain Tuning PSO PID dengan parameter W = 0.9, C1 = 1, C2 = 0.5 29](#_Toc121988171)

[29](#_Toc121988172)

[4.5 Implementasi Simulasi Tuning PID dengan PSO pada Sistem Pengaturan Cascade Level dan Flow 30](#_Toc121988173)

[4.6 Pembahasan 30](#_Toc121988174)

[BAB 5 Kesimpulan dan Saran 31](#_Toc121988175)

[5.1 Kesimpulan 31](#_Toc121988176)

[5.2 Saran 31](#_Toc121988177)

[DAFTAR PUSTAKA 32](#_Toc121988178)

[LAMPIRAN 34](#_Toc121988179)

[BIODATA PENULIS 35](#_Toc121988180)

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SIMBOL

# PENDAHULUAN

## Latar Belakang

Dalam sistem kontrol proses industri skala besar, variabel yang sering dikontrol adalah level cairan, laju aliran, suhu, tekanan dan sebagainya, yang sebagian besar merupakan fungsi non linier dari input kontrol. Namun, untuk menggunakan teknik kontrol linear secara umum proses nonlinier perlu didekati untuk mendapatkan model linierisasi, salah satunya adalah fungsi alih. (Wen et al., 2012). Masalah utama dalam industri pengolahan adalah pengaturan level dan aliran cairan. Saat ini, sering terjadi masalah dengan alat kontrol pada tangki reaktor selama proses pengaturan level dan aliran. Contohnya, kerusakan pada instrumen pengukur dan gangguan tak terduga. Hal ini tentu saja mempengaruhi operasi sistem kontrol level dan aliran cairan. Untuk mencapai kinerja yang baik dalam sistem pengaturan level dan aliran, diperlukan pengontrol yang mampu menangani masalah yang muncul dalam sistem, yaitu kontrolir PID. (Pritandi, 2016). Kontrolir proporsional-integral-derivatif (PID), yang merupakan algoritma kontrol *feedback*, telah banyak diterapkan dalam kontrol industri karena kinerja yang robust dalam berbagai kondisi operasi dan kemudahan fungsional. Lebih dari 90% aplikasi kontrol industri didasarkan pada mekanisme kontrol PID, seperti sistem robotika dan pengendalian otomatis. (Xiang et al., 2019)

Namun dalam system pengaturan yang terdapat controller PID masih belum optimal terutama adanya disturbance atau gangguan berupa ketidakstabilan laju aliran pada variable yang dimanipulasi sehingga output juga berubah. Kontrol kaskade adalah salah satu metode paling sukses untuk meningkatkan kinerja kontrol loop tunggal dengan menstabilkan, mengurangi osilasi dan kesalahan integral dari gangguan pada variabel yang dimanipulasi (Tridianto et al., 2017). Sistem pengaturan kaskade pada umumnya masih menggunakan controller PID konvensional pada industri proses terutama untuk mengatur level dan aliran. Kelemahan dari pengontrol konvensional tidak dapat diabaikan karena sebagian besar industri berfokus pada akurasi, margin keuntungan, dan waktu operasi. (Guong et al., 2018). Terinspirasi oleh perilaku sosial kawanan burung dan kelompok ikan, PSO diusulkan sebagai metaheuristik berbasis populasi teknik optimasi untuk solusi dari berbagai fungsi nonlinier dan masalah optimisasi. Sudah beberapa tahun, PSO terbukti cukup populer di berbagai bidang terutama karena sederhana, *Robustness*, dan *high dimensionality* (Pano & Ouyang, 2014). Terdapat penelitian yang membahas perihal *Particle Swarm Optimization* diantaranya adalah PSO digunakan untuk desain yang optimal pengontrol PID untuk gerakan DC brushless linier(Ibrahim et al., 2014),dan Pengendalian PID multi-objektif Berdasarkan PSO Dengan aplikasi ke steam Pengendalian suhu di Boiler (Kumar et al., 2010).

## Rumusan Masalah

Pada industri proses sering dijumpai system pengaturan kaskade dimana salah satunya adalah Sistem Pengaturan Kaskade Level dan *Flow* atau aliran. Sistem pengaturan kaskade terdiri dari inner loop dan outer loop dimana masing-masing terdapat kontroler PID, karena Sebagian besar industri sekarang berfokus pada akurasi, margin keuntungan dan waktu operasi maka tuning PID konvensional belum optimal dalam menyelesaikan masalah tersebut. Oleh karena itu diperlukan cara tuning PID dengan Algoritma PSO untuk mendapatkan hasil yang optimal. Rumusan masalah pada Tugas Akhir ini adalah bagaimana cara menala controller PID dengan algoritma Particle Swarm Optimization dan Bagaimana respons system setelah *detuning* dengan Algoritma PSO tersebut.

## Batasan Masalah

Dalam melaksanakan Tugas Akhir ini, terdapat berbagai batasan masalah yang perlu diperhatikan, diantaranya sebagai berikut:

1. *Plant* yang digunakan pada Tugas Akhir ini adalah PCT 100 dengan system yang digunakan adalah Sistem Pengaturan Level dan Sistem Pengaturan Flow yang selanjutnya dimodelkan persamaan model linear fungsi alih.
2. Implementasi simulasi dan pengujian dengan menggunakan simulator MATLAB dan SIMULINK.
3. Parameter yang digunakan pada Particle Swarm Optimization dibuat tetap.

## Tujuan

Tujuan dari tugas akhir ini adalah:

1. Merancang algoritma tuning PID dengan metode particle swarm optimization sehingga di dapat nilai optimal dari masing-masing controller Proportional, Integral dan Derivatif.
2. Mengetahui Respons system setelah dituning dengan PID controller menggunakan metode particle swarm optimization serta dianalisa karakteristik respons sistem

## Manfaat

Manfaat penelitian ini diharapkan tuning PID berbasi Particle Swarm Optimization pada sistem pengaturan kaskade level dan flow dapat menjadi solusi optimal pada permasalahan tuning PID konvensional dan dapat diaplikasikan dalam industri control proses, serta sebagai referensi pada penelitian selanjutnya.

**(*halaman ini sengaja dikosongkan*)**

# TINJAUAN PUSTAKA

## Hasil Penelitian Terdahulu

## Plant PCT 100

*Process Control Technology* atau biasa disebut *PCT-100* adalah sebuah *plant* mewakili suatu sistem yang biasanya ditemukan di industri proses. Pada plant ini terdiri dari Modul Proses, Konsol Kontrol dengan Catu Daya, dan perangkat lunak kontrol interaktif.



Konsol Kontrol merupakan representasi dari modul proses yang terletak pada bagian depan dan terdapat *fault switch* dan titik uji. Konsol Kontrol terdapat koneksi untuk memungkinkan kontrol pada *Personal Computer* dengan menggunakan kabel (USB) dan PLC atau pengontrol PID. Pada plant PCT 100 terdapat cairan yang terkandung dalam *sump tank* dipompa di sekitar sistem ke dalam tangki proses di mana level, temperatur dan tekanan dapat dikontrol. Dalam tangki proses, level diukur menggunakan sensor magnetostriktif 0-10 Volt; tekanan diukur menggunakan sensor gauge 0-5 bar; dan PT1000 digunakan untuk mengukur suhu di tangki bah dan tangki proses. Sensor laju aliran turbin digunakan untuk mengukur laju aliran dalam sistem. Katup pengalih dapat digunakan untuk mengarahkan cairan melalui proses pendinginan udara untuk mendinginkan cairan dalam sistem. Dua katup proporsional digunakan untuk mengontrol aliran masuk dan keluar dari tangki proses, katup jarum yang dapat disesuaikan secara manual digunakan untuk menambah gangguan pada sistem dan katup pelepas tekanan yang dipasang untuk keselamatan.

Data yang dikumpulkan dapat ditampilkan pada LCD yang dipasang pada Modul Proses dan melalui perangkat lunak data dapat dipantau, disimpan maupun dicetak. Perangkat lunak pada modul ini digunakan sebagai modul untuk belajar kontroler PID dan menyediakan fitur *monitoring* serta akuisisi data dengan beberapa fitur yang sedang tren.

Pada PCT-100 ini terdapat modul untuk pengaturan dan eksekusi siklus proses, hingga optimalisasi pengontrol PID dengan memperkirakan parameter P, PI, dan PID menggunakan kurva reaksi proses Ziegler Nichols atau metode siklus akhir.(Bytronic Limited, n.d.)

Tabel 2‑1 Tabel Spesifikasi Plant PCT 100

|  |  |
| --- | --- |
| **Specification** | |
| **Process Module** |  |
| Sump tank volume | 8 L |
| Process tank volume | 4.5 L |
| Heating element power | 48v dc 400W |
| Control elements | 2 x Proportional valves |
| 1 x Manually adjustable needle valve |
| Pump | 24v 0.5amp 6 Litres per minute with 1.5 bar safety cut-out |
| Cooling system | Radiator with forced air cooling |
| Maximum flow rate | 2.7 Litres per minute |
| Level transducer | Magnetostrictive position sensor 0 – 10v |
| Pressure transducer | Gage 0 – 5 bar (0 – 2 bar is used as the range for laboratory exercises) |
| Number of LCD displays | 5 x level, process tank and sump tank temperature, pressure and flow rate |
| Indicators | 4 x heater on, drain valve open, diverter solenoid active and cooler active |
| **Control Console** |  |
| Number of fault switches | 6 x Illuminated |
| Test Points | 5 x heater, level, temp; sump tank and process tank, flow rate |
| Indicators | 6 x heater, drain , flow and diverter solenoid, pump and cooler active |
|  | Graphical representation of Process Module |
| Connections | 1 x 15 way ‘D ‘connector and 1 x 25 way ‘D’ connector |
|  | 1 x power connector for heater |
|  | 1 x USB connection |
|  | 1 x Mains power connector |
| Actuator voltage | 24Vdc |
| Power Supply voltage | 100-250V AC @ 50/60Hz. |
| Approximate weight | 42Kg |
| Approximate dimensions | 100x76x40cm |
| **Software** | Windows based software with SCADA type interface. PID and Manual Control  of Flow, Level. Temperature, Pressure, Open Loop and Batch. Data logging with printing and saving features |

## Pemodelan *Plant* PCT-100

Pada PCT-100 *plant* yang digunakan pada Gambar (Ket) menggambarkan proses pada sistem pengaturan level atau ketinggian suatu fluida. Adapun proses pengaturan level berupa pengisian tangka reaksi yang dilakukan oleh pompa sedangkan pembuangan fluida dilakukan oleh kontrol *valve* dan variabel *valve* pada PCT-100, kedua proses tersebut jika digabungkan maka akan mengontrol luaran pada sistem yakni sistem pengaturan level.



Proses pengaturan level digambarkan dengan menggunakan diagram alir pada Gambar (Ket). Pemodelan ini dimulai dari pemodelan pompa, pemodelan pipa, pemodelan tangka, dan pemodelan sensor. Adapun diagram alirnya adalah:

Diagram

Description automatically generated

Parameter yang diperlukan dalam pemodelan *plant* terdaoat pada pada Tabel (ket) beserta keterangan dan satuan yang digunakan nilainya diambil dari *data sheet* dan pengukuran langsung pada *plant* PCT-100.

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Keterangan** | **Satuan** |
| Qin | Debit air yang masuk ke dalam tangka | *cm3 / s* |
| Qout | Debit ari yang keluar dari tangka | *cm3 / s* |
| Q | Debit air dalam tangka | *cm3 / s* |
| H | Ketinggian air dalam tangki | *Cm* |
| A | Luas alas tanki | *cm2* |
| a | Diameter pipa | *cm* |
| h0 | Ketinggian air awal pada saat titik operasi | *cm* |
| Q0 | Laju aliran keluar saat kondisi di titik operasi | *cm3 / s* |
| R | Resistansi *control valve* | *s / cm* |
| C | Kapaistansi tangka | *cm2* |
| L | Panjang pipa | *cm* |
|  | Waktu tunda | Detik |

Untuk memperoleh model *plant* PCT 100 terutama sistem pengaturan level maka dalam prosesnya harus digunakan persamaan atau formula untuk memodelkan Pompa, Pipa, Tangki dan Sensor adapun formulanya adalah sebagai berikut.

### Pemodelan Pompa

Pada pemodelan pompa ini bertujuan untuk mendapatkan nilai penguatan dari pompa yang dilambangkan dengan K1 serta dirumuskan melalui persamaan (Ket)

### Pemodelan Pipa

Tujuan dalam pemodelan pipa ini adalah untuk mengetahui besarnya waktu tunda yang diakibatkan oleh panjang pipa. Hubungan waktu tunda dengan panjang pipa dapat dirumuskan dalam persamaan berikut :(ket)

Persamaan di atas dapat dibawa ke sistem pengaturan level dimana hubungan antara debit air yang dikeluarkan oleh pompa dan debit air yang masuk ke tangki adalah sebagai berikut

### Pemodelan Tanki

Pada sebuah tangki reaksi sistem pengaturan level proses yang terjadi pada umumnya terdiri dari pengumpanan dan pembuangan air. Proses pengumpanan bertujuan untuk menambah volume air sedangkan proses pembuangan akan mengurangi volume air pada tangka reaksi. Parameter utama yang digunakan dalam sistem reservoir air berupa kapasitansi tanki dan resistansi jalur pembuangan. Kapasitansi reservoir didefinisikan sebagai besar perubahan volume cairan yang diperlukan untuk membuat perubahan ketinggian sebesar satu satuan(Ogata, 2010). Kapasitansi suatu tangki atau reservoir sama dengan luas permukaan penampang tangki atau reservoir tersebut. Jika luas penampang konstan maka nilai kapasitansi juga konstan

Berdasarkan asumsi tersebut maka persamaan kesetimbangan massa adalah sebagai berikut:

Dengan demikian sistem tangki air sebenarnya merupakan sistem yang tidak linier. Pada umumnya linierisasi dapat dilakukan untuk sistem pengendalian level air dengan daerah kerja ketinggian level yang sudah ditetapkan, sehingga linierisasi nilai resistansi dapat dilakukan (Kawasan et al., 2011). Metode linierisasi ditunjukkan pada gambar (ket) dibawah ini

Chart

Description automatically generated

Untuk melakukan linearisasi berdasarkan grafik di atas maka diperoleh persamaan yang dinyatakan sebagai berikut

Dengan demikian rumus kesetimbangan massa pada persamaan (ket) menjadi :

Dari persamaan di atas dapat ditransformasikan Laplace sehingga diperoleh persamaan berikut:

Dan diperoleh fungsi alih sistem yakni:

Dengan adalah konstanta waktu sistem tanki, *Qi* adalah debit masukan air (cm3/s), *Qo* adalah debit luaran air (cm3/s), H adalah level air (cm), dan g adalah percepatan gravitasi sedangkan A0 adalah luas penampang pipa keluaran.

### Pemodelan Sensor

Pemodelan sensor bertujuan untuk mendapatkan nilai penguatan yang dilambangkan dengan K2. Nilai K2 dirumuskan dengan rentang kerja dari pompa terhadap masukan berupa tegangan seperti Persamaan (ket) dibawah ini

Dari uraian persamaan pemodelan di atas didapat persamaan *plant* sistem pengaturan level seperti berikut:

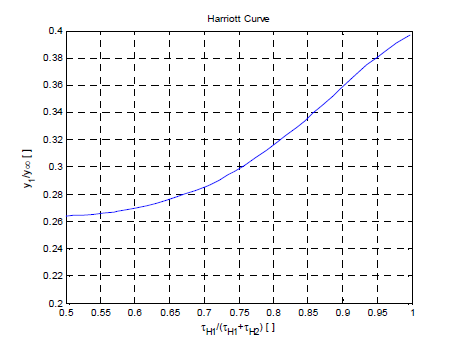
## Pemodelan Sistem Dengan Menggunakan Kurva Reaksi

Metode kurva reaksi adalah sebuah metode yang digunakan untuk mengidentifikasi model matematis sebuah sistem dinamik. Metode ini mengandalkan pengukuran kurva respons sistem terhadap suatu sinyal *input* yang diketahui. Dengan menggunakan metode ini, sistem dinamik dapat diidentifikasi sebagai suatu model matematis seperti persamaan diferensial atau sistem persamaan linear.

Metode Harriott (Jakoubek, 2009) mencoba untuk mensimulasikan sistem dengan menggunakan model orde 2 yang memungkinkan untuk menyertakan delay waktu. Fungsi transfer model yang dihasilkan dapat dituliskan sebagai berikut:

Chart, line chart

Description automatically generated



Berikut adalah tahapan untuk mendapatkan parameter , dan :

### Menentukan jumlah dari parameter dan

Jumlah dari parameter dan dapat dirumuskan sebagai berikut:

Dengan adalah waktu saat respons system berada pada kondisi 73% mendekati luaran *steady state* ().

### Menentukan parameter dan

Parameter parameter diperlukan untuk menghitung parameter, yang diperoleh dari respon sistem yang telah diidentifikasi. Rumus untuk menghitung parameter adalah sebagai berikut:

### Menentukan parameter pada kurva Harriot

Pada kurva Harriott, parameter yang diberikan terletak pada sumbu X dan Y. Parameter yang ditunjukkan pada sumbu Y adalah sebagai berikut:

Setelah menentukan parameter pada sumbu Y, garis dapat ditarik untuk mendapatkan parameter / pada sumbu X. Parameter ini merupakan koordinat sumbu X pada kurva Harriott. Dengan demikian, kita dapat menggambarkan kurva Harriott dengan menggunakan parameter yang diperoleh pada kedua sumbu tersebut.

### Menentukan parameter

Untuk mencari parameter maka persamaan nilai / dapat diasumsikam pada kurva Harriot sebagai x, kemudian substitusikan persamaan ()

### Menentukan Parameter

Untuk mendapatkan parameter , subsitusikan persamaan 2.24 pada persamaan 2.20, sehingga diperoleh:

### Menentukan parameter

adalah *time delay* yang dapat di metode Harriot dirumuskan sebagai berikut:

Untuk menggunakan hasil parameter , , dan Parameter tersebut dapat memasukkannya ke dalam persamaan 2.19. Jika nilai waktu dan kurang dari 0 atau bernilai negatif, maka sistem dianggap tidak memiliki time delay. Dengan menggunakan hasil parameter tersebut dalam persamaan 2.19, sehingga dapat ditentukan kinerja sistem tersebut.

## Kontroler PID

Algoritma Kontroler PID merupakan kontrolir yang paling popular pada proses industri, meskipun ada kemajuan terus menerus pada teori kontrol. Hal ini dikarenakan algoritma PID memiliki struktur yang sederhana dan mudah dipahami secara konseptual serta dalam praktiknya kontroler ini memberikan respons kerja yang sangat memadai dalam aplikasinya pada *plant* di industri. Kontrolir PID bisa dianggap sebagai bentuk “ekstrem’ dari kompensator lead-lag dengan fase satu kutub di titik asal dan yang lainnya pada titik tak terhingga. Adapun bentuk kontrolir PID ditulis dalam bentuk persamaan paralel di mana (Xiong & Fan, 2007)

Di mana 𝐾𝑃 adalah *gain proportional*, 𝐾𝑖 adalah *gain integral*, 𝐾𝑑 adalah *gain derivative*, 𝑇𝑖 adalah konstanta *integral time*, dan 𝑇𝑑 adalah konstanta *derivative time.*

Untuk mendapatkan hasil tuning kontroler PID yang maksimal pada umumnya digunakana aturan tuning yakni *Ziegler-Nichols* atau *Coheen Coon*. Adapun karakteristik kontroler PID (Ang et al., 2005)adalah sebagai berikut

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Respon *Closed Loop*** | ***Rise Time*** | ***Overshoot*** | ***Settling time*** | ***Steady state error*** | **Stabilitas** |
| Menaikkan Kp | Berkurang | Mengalami perubahan | Sedikit mengalami kenaikan | Mengalami penurunan | Menurunkan |
| Menaikkan Ki | Berkurang | Bertambah | Meningkat | Menurunkan signifikan | Menurunkan |
| Menaikkan Kd | Sedikit Perubahan | Berkurang | Berkurang | Berdampak minor | Menaikkan |

## Sistem Pengaturan *Cascade*

Sistem pengaturan *cascade* adalah sistem yang terdiri dari inner loop dan outer loop seperti pada gambar (ket). di mana *inner loop* berasal dari pengenalan sensor tambahan untuk memisahkan sebanyak mungkin dinamika sistem proses cepat dan lambat. Akibatnya, *control* konfigurasi sistem memiliki pengontrol bagian dalam *C2(s)* dengan bagian dalam proses loop *G2(s)* dan kontroler *loop* luar *C1(s)* dengan outer proses *loop* *G1(s*). Gangguan bisa masuk di dua kemungkinan titik yang berbeda: *d1* dan *d2*. Alasan di balik konfigurasi ini adalah untuk dapat mengkompensasi yang terbaik, yang mungkin gangguan *d2*, sebelum direfleksikan ke *output loop* luar. Untuk mencapai tujuan atau *set point* yang diinginkan adalah penting bahwa *inner* *loop* menunjukkan dinamika yang lebih cepat yang memungkinkan untuk kompensasi.

Diagram, schematic

Description automatically generated

Berdasarkan uraian di atas proses *G(s) = G1(s)G2(s)* dibagi menjadi dua bagian *G1(s)* dan *G2(s)* dan yang terkait pengontrol. Kedua pengontrol tersebut adalah pengontrol umpan balik standar yang diasumsikan mengambil bentuk PID biasa yakni:

Dimana *Kp1* dan *Kp2* adalah *gain* proporsional, *Ti1* dan *Ti2*adalah *gain* integral, *Td1*dan *Td2* *gain* turunan dan serta *N1* dan *N2* konstanta filter *noise* atau gangguan waktu *derivative*.(Vilanova & Arrieta, 2008)

## *Particle Swarm Optimization*

Pendekatan PSO menggunakan algoritma optimisasi stokastik berbasis populasi yang diusulkan oleh Eberhart dan Kennedy pada tahun 1995. Hal tersebut terinspirasi dari simulasi komputer yang mengambil contoh dari perilaku sosial kawanan burung, simulasi tersebut dicetuskan oleh Reynolds pada tahun 1987. Reynolds menggunakan grafik komputer untuk memodelkan perilaku berkelompok burung yang rumit. Dia tertarik pada simulasi pola terbang burung untuk tujuan simulasi komputer visual, mengamati bahwa kawanan tampaknya berada di bawah kendali terpusat. Reynolds mulai memodelkan penemuannya menggunakan tiga aturan sederhana, yaitu penghindaran tabrakan, pencocokan kecepatan, dan pemusatan kawanan. Dengan menggunakan aturan ini, Reynolds menunjukkan bagaimana perilaku masing-masing agen di dalam flok dimodelkan dengan vektor sederhana. Karakteristik ini merupakan salah satu konsep dasar PSO (Pillay, 2013).

PSO adalah metode pencarian heuristik yang terinspirasi oleh interaksi sosial dari hewan yang hidup berkelompok. PSO dimulai dengan sekelompok partikel yang dihasilkan secara acak, yang masing-masing merupakan solusi yang mungkin untuk masalah optimasi. Setiap partikel diberi nilai posisi, dievaluasi oleh *fitness fuction* dirancang untuk mewakili masalah yang akan dioptimalkan. Selanjutnya, setiap partikel memiliki kecepatannya sendiri yang menentukan gerak partikel dalam ruang solusi (Pano & Ouyang, 2014).

Berdasarkan uraian tersebut maka dapat dinyatakan dalam bentuk persamaan yakni :

Di mana adalah kecepatan sekarang agen i pada iterasi k, adalah pembaruan kecepatan agen i pada iterasi k, 𝑐1 dan 𝑐2 merupakan faktor *learning, rand* adalah factor random yang terdistribusi pada [0,1], 𝑝𝑏𝑒𝑠𝑡 merupakan solusi optimal suatu partikel, dan 𝑔𝑏𝑒𝑠𝑡 merupakan solusi optimal seluruh populasi, sementara itu *s* merupakan lokasi partikel dan *𝑣* merupakan kecepatan partikel.

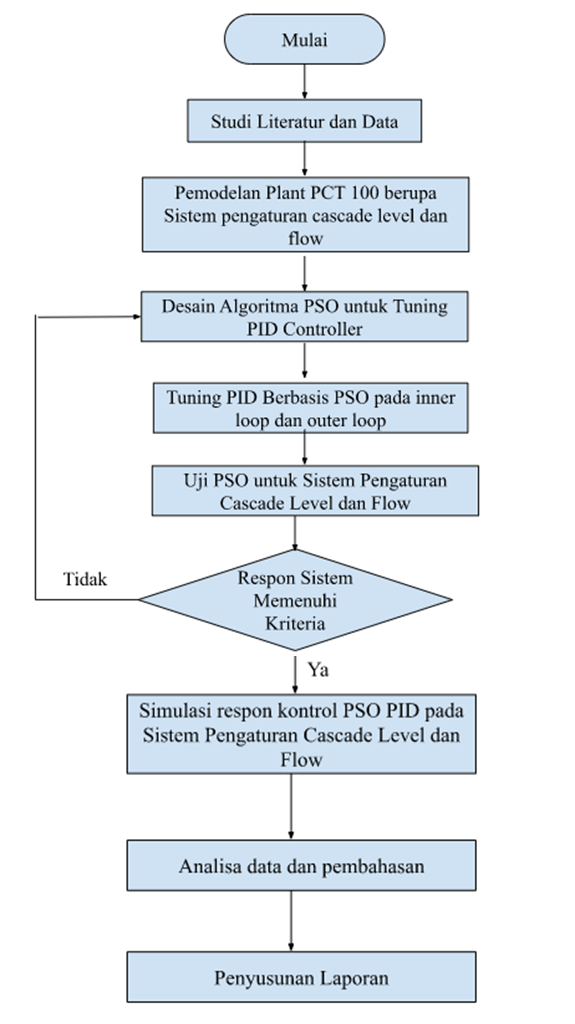
Diagram

Description automatically generated

# METODOLOGI

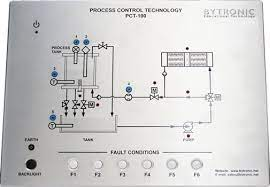
## Metode yang digunakan

Dalam tugas akhir ini, metode utama yang digunakan yaitu simulasi. Pada simulasi, terdapat beberapa langkah khusus akan dilakukan, mulai dari identifikasi sistem, pemodelan sistem, perancangan kontroler, hingga pengambilan data simulasi. Metode identifikasi sistem dalam simulasi akan dilakukan dengan memberikan sinyal uji pada flow plant PCT 100 sehingga diketahui karakteristik dari dari *flow plant* tersebut. Sedangkan untuk sistem pengaturan level akan dilakukan *modelling* dengan matematis berdasarkan parameter level tanki pada PCT 100. Perancangan controller akan dilakukan dengan mengaplikasikan algoritma PSO mencari parameter kontroler PID serta dibandingkan dengan tuning PID Ziegler Nichols. Adapun metodologi pada penelitian ini adalah:



## Bahan dan peralatan yang digunakan

### Process Control Technology 100 Byotronics



PCT-100 adalah *miniplant* salah satu produk dari *Bytronic* yang digunakan untuk mendemonstrasikan prinsip kontrol dalam proses instrumentasi dan sistem kontrol proses. Pada peralatan ini terdapat proses yang terjadi pada tangki utama, yakni ada empat variabel proses yang diatur yaitu tekanan *(pressure*), level air, aliran air *(flow*), dan suhu (temperature). Aktuator yang terdapat pada PCT -100 adalah pompa dan pemanas. Pompa digunakan untuk menyalurkan air dari tangki penyimpanan menuju tangki utama. Sementara pemanas digunakan dalam proses sistem pengaturan suhu. Keempat variabel tersebut diatur dengan menggunakan kontrolir PID pada program bawaan alat ini

### Adam 5000 L TCP/IP



ADAM-5000L / TCP dan ADAM-5000 / TPC adalah sistem I / O berbasis Ethernet yang dirancang untuk konfigurasi yang mudah dan pengelolaan yang efisien. Mereka dapat digunakan untuk mengkonfigurasi dan memantau data secara jarak jauh dari berbagai perangkat, seperti sensor dan aktuator, melalui jaringan Ethernet. Kedua model memiliki jarak komunikasi hingga 100 m tanpa perlu repeater, dan hingga delapan PC dapat secara bersamaan mengakses data. Ini menjadikan mereka solusi yang efektif dan fleksibel untuk berbagai aplikasi otomasi.

Pada Tugas Akhir ini ADAM 5000L digunakan sebagai Input Ouput Module dikarenakan kecocokan pada sistem. Fungsi dari ADAM sendiri untuk memberi sinyal perintah berupa tegangan kepada Pompa PCT 100 dan pembacaan sensor flow rate dari *plant* PCT 100 dan juga sebagai penerima sinyal dari sensor yang berupa tegangan. Spesifikasi dari ADAM 5000L dapat dilihat dari Gambar Tabel

|  |  |
| --- | --- |
| Konsumsi Daya | 4W, 24 VDC (tdaik termasuk modul I/O) |
| Power Input | 10-30 VDC |
| Konektor | 1 x DB9-M/DB9-F/screw terminal untuk RS-485 (komunikasi), 1 x DB9- F untuk RS-232 (penggunaan internal), 1 x Screw terminal untuk power input, 1 x RJ-45 untuk LAN |

### Matlab



MATLAB adalah singkatan dari Matrix Laboratory dan merupakan sebuah bahasa pemrograman dan lingkungan komputasi numerikal yang digunakan untuk mengelola data, menghitung, dan mengembangkan algoritma. MATLAB dikembangkan oleh MathWorks dan menyediakan fungsi-fungsi matematis, visualisasi data, dan pemodelan integratif yang kuat untuk menganalisis dan memvisualisasikan data serta membangun aplikasi yang didasarkan pada hasil analisis tersebut. MATLAB juga dapat digunakan untuk membangun sistem kontrol, memproses sinyal, dan menganalisis data secara real-time.

### Simulink



Simulink adalah sebuah perangkat lunak yang dikembangkan oleh MathWorks yang memungkinkan untuk menggambar dan mengubah model sistem dinamis dalam bentuk diagram blok. Simulink memiliki banyak library yang berisi blok-blok yang mewakili fungsi-fungsi yang biasa digunakan dalam pemodelan sistem dinamis, seperti filter, sumber, integrator, dan lainnya. Simulink terdapat bagian blok diagram dan dan antar blok dapat dihubungkan untuk membangun model sistem yang sesuai dengan spesifikasi yang diinginkan. Kemudian, Simulink digunakan untuk menganalisis dan menjalankan simulasi model tersebut untuk menguji dan mengevaluasi bagaimana sistem akan berperilaku dalam situasi nyata.

### Personal Computer

Komputer yang digunakan pada Tugas Akhir ini memiliki spesifikasi sebagai berikut

|  |  |
| --- | --- |
| Prosesor | Intel(R) Core(TM) i5-8250U |
| RAM | 12GB |
| STORAGE | 1TB SATA (5400 rpm) |

## Urutan pelaksanaan penelitian

### Pemodelan Sistem Pengaturan Level pada PCT 100

#### Pemodelan Pompa

Mengacu pada Persamaan () dengan memasukkan rentang debit air yang keluar dari pompa antara 0 - 45 cm/s . Rentang tegangan masukan antara 0 – 10 V didapatkan hasil pada Persamaan ()

#### Pemodelan Pipa

Mengacu pada Persaman () dengan memasukkan data diameter pipa sebesar 1 cm, panjang pipa sebesar 90 cm dan dengan debit air yang masuk 16.67 cm3/detik didapatkan Persamaan()

Diperoleh *time delay*:

#### Pemodelan Tangki

Nilai Q0 didapatkan dengan melakukan uji pengosongan tangki. Diambil titik kerja ketika nilai h0 sebesar 7.5 cm didapatkan nilai Q0 seperti pada Persamaan (ket)

V adalah volume air yang keluar dan t adalah waktu yang dibutuhkan untuk mengosongkan tangki. Mengacu pada Persamaan (ket) dan Persamaan (ket) didapatkan nilai R dan C pada Persamaan (ket) dan (ket)

Sehingga di dapat fungsi alih:

#### Pemodelan Sensor

Mengacu pada Persamaan (Ket) dengan memasukkan rentang tegangan keluaran antara 0 – 10 V dan rentang ketinggian air dalam tangka antara 0 – 18 cm data sesuai dengan user manual PCT-100 , maka penguatan sensor dirumuskan pada Persamaan (ket)

#### Persamaan Model Sistem Pengaturan Level Total

Mengacu pada Persamaan () dapat dihitung fungsi alih keseluruhan dari plant dengan cara mengalikan semua komponen pemodelan seperti pada Persamaan ()

### Pemodelan Sistem Pengaturan Flow

Dalam proses mengembangkan dan mendesain kontrolir untuk suatu plant, pengetahuan tentang sifat dan karakteristik plant tersebut sangat penting. Salah satu cara untuk mengetahui sifat dan karakteristik *plan*t adalah dengan menganalisis fungsi alih yang dimiliki oleh plant tersebut. Dengan mengetahui fungsi alih ini, kita dapat memahami karakteristik dari plant tersebut dan mengembangkan kontrolir yang sesuai untuk mengendalikannya. Salah satu permasalahan yang sering muncul ketika menghadapi suatu *plant* adalah tidak adanya dokumen yang menyediakan informasi tentang fungsi alih dari plant tersebut. Tanpa informasi ini, sulit bagi para peneliti untuk memahami karakteristik plant dan mengembangkan kontrolir yang sesuai untuk mengendalikannya. Oleh karena itu, penting bagi para peneliti untuk mencari tahu informasi ini sebelum mulai mengembangkan kontrolir untuk suatu *plant*.

Proses identifikasi adalah suatu cara untuk menentukan nilai parameter-parameter dari suatu plant dengan menganalisis respon plant terhadap sinyal uji yang diberikan. Proses ini biasanya dilakukan secara *open loop*, di mana sinyal uji diberikan dari komputer kontrolir ke *plant*, dan respon dari plant terhadap sinyal uji tersebut diterima kembali oleh komputer kontrolir. Kemudian, keluaran dari plant diolah menjadi sebuah grafik, dan dari grafik tersebut kita bisa mendapatkan nilai parameter-parameter plant. Identifikasi biasanya dilakukan pada kondisi *set point* yang berbeda-beda.

Menurut jurnal dengan penulis Ing. Pavel Jakoubek, terdapat 7 metode identifikasi yang dapat digunakan untuk sistem tanpa osilasi dengan masukan respons step. Setelah diperoleh fungsi alih sistem, ketujuh metode ini dibandingkan dengan respons hasil pengukuran untuk menentukan metode dengan pendekatan model yang terbaik. Metode validasi yang biasa digunakan dalam kasus ini adalah *ISE (Integral Square Error*). Dengan menggunakan metode ini, kita dapat membandingkan hasil dari model yang diperoleh dengan data yang diukur dari *plant* untuk menentukan metode yang paling baik untuk digunakan.

Untuk memodelkan flow kita menggunakan Adam Advantech sebagai jembatan untuk memperoleh data, Adam Advantech juga berfungsi sebagai ADC maupun DAC yang berguna untuk mengubah sinyal analog maupun digital atau converter tentunya berguna dalam control kontinu maupun diskrit. Adapun diagaram pemodelan plant flow adalah sebagai berikut:

A picture containing text

Description automatically generated

#### Pengambilan data Plant PCT 100 Sistem Pengaturan Flow

Untuk mendapatkan fungsi alih dari sistem dirancang sistem *closed loop* pada gambar () yang terdiri personal computer, ADAM 5000 L TCP/IP, dan Plant PCT 100. Setelah itu digunakan input berupa tegangan 5 Volt pada ADAM 5000L TCP/IP untuk memperoleh respons sistem yang akan dibaca oleh sensor flow. Digunakan algoritma 1 berikut ini untuk membaca hasil output dari sensor.

|  |  |
| --- | --- |
| **ALGORITME 1. Algoritme Pemodelan Plant Flow pada PCT 100** | |
|  | **Input**: Tegangan DC 5 Volt |
|  | **Output** : Bacaan sensor basis uint16 |
| 1. | ***Data* :** Plot Grafik uint16 (1:0.01:30) |
| 2. | ***Set* : Alamat IP 169.254.126.231 pada port 502 menggunakan protokol Modbus** |
| 3. | ***Set* : Timeout untuk koneksi Modbus 30 detik** |
| 4. | ***Set* : ID server = 1** |
| 5. | ***Write : nilai 0 menggunakan fungsi “holdingregs” dengan alamat PORT = 17*** |
| 6. | ***Read : Baca data dari” holdingregs” PORT = 9 dan simpan sebagai "data"*** |
| 7. | ***Set : variable waktu t = 0*** |
| 8. | ***For*** |
|  | Baca data dari “holdingregs” PORT = 9 dan tambahkan ke "data"  Tambahkan "t" dengan waktu yang telah berlalu sejak iterasi sebelumnya  Plot "t" dan "data"  Gambarkan plot pada layar  Jeda selama 0,01 detik |
| 9. | ***end*** |
| 10. | ***Putuskan koneksi dari perangkat*** |

Berdasarkan algoritma tersebut kemudian diimplementasikan pada Matlab untuk memperoleh data dari sensor *Flow Rate* diperoleh respons sistem berikut pada gambar ()

Graphical user interface

Description automatically generated

Dari respons sistem tersebut masih dalam berbentuk bilangan bit yakni uint16 untuk itu perlu dilakukan konversi ke bilangan asli pembacaan sensor sesuai spesifikasi sensor *flow rate* pada PCT 100. Bilangan uint 16 sendiri adalah bilangan dengan panjang 16 bit dimana dimulai dari 0-6555. Untuk respons sistem yang dihasilkan angka 0 liter/menit dimulai dari bilangan bit sebesar 32768 dan 2.7 liter/menit diangka 65535. Oleh karena itu dilakukan konversi dengan algoritma sebagai berikut:

|  |  |
| --- | --- |
| **ALGORITME 2. Algoritme Konversi bilangan uin16 ke bilangan basis sensor** | |
|  | ***Set :*** uint16\_value = 32768; |
|  | // Konversi bilangan uint16 ke bilangan satuan sensor 2.7 liter/menit  ***Hitung*** sensor\_value = (uint16\_value / 65535) \* 2.7; |
|  | // Cetak hasil konversi  **Print**(sensor\_value); |
|  | // Baca file CSV  **data** = ReadCSV('nama\_file.csv'); |
|  | // Konversi nilai dalam matriks menjadi bilangan satuan sensor 2.7 liter/menit  converted\_data = **Convert**(data, @(x) (x / 65535) \* 2.7); |
|  | // Buat plot grafik dari hasil konversi  **Plot**(converted\_data); |

Dari algoritma tersebut diperoleh grafik respons system sebagai berikut pada gambar (). Dimana dari respons system tersebut akan dianalisa untuk dicari model fungsi alih dari system.

Graphical user interface

Description automatically generated with medium confidence

#### Fungsi Alih Sistem

Untuk mendapatkan fungsi alih sistem, digunakan respon saat set point adalah 5 volt atau 50% dari flow rate Gambar 3.4 menunjukkan respon level pada set point 50%. Dari respon tersebut, diperoleh nilai keluaran steady state pada ketinggian 1,939, yang diperoleh dari rata-rata seluruh data keluaran steady state. Berdasarkan respon pada Gambar 3.4, diperoleh gain overall respon pada persamaan.

Jakoubek mengembangkan 7 metode untuk mengidentifikasi plant atau sistem kontrol. Salah satu metode yang dapat digunakan dalam pendekatan model matematika plant adalah metode Harriott. Pemilihan metode terbaik dapat dilakukan dengan menggunakan validasi. Salah satu metode validasi yang dapat digunakan adalah ISE (Integral Square Error). Semakin kecil nilai ISE, maka semakin baik fungsi alih yang dibuat. Dengan menggunakan pendekatan sistem model matematika plant proses level dan berdasarkan respon pada Gambar 3.4, dapat diperoleh nilai ISE yang sesuai

Sehingga diperoleh parameter:

Berdasarkan kurva Harriott pada Gambar 2.5, diperoleh nilai:

Karena parameter Parameter delay bernilai negatif, sehingga nilai delay tidak dianggap. Fungsi alih untuk metode Harriot adalah:

Penelitian ini mengukur respon dari model terhadap masukan sinyal uji untuk menentukan kesesuaiannya dengan *real plant* dalam proses identifikasi. Hasil percobaan menunjukkan bahwa ISE sebesar dan model matematikanya merupakan pendekatan orde 2, yang umumnya sesuai dengan sistem orde 2 pada dunia industri.

### Tuning PID Menggunakan pada Sistem Pengaturan Cascade

*Cascade control* adalah metode kontrol yang menggunakan dua atau lebih PID (proportional-integral-derivative) controller untuk mengendalikan suatu sistem. Dalam kontrol kaskade, salah satu *PID controller* disebut "kontrolir primer" dan yang lainnya disebut "kontrolir sekunder". Dalam tugas akhir ini dirancang sebuah system cascade pada matlab yang terdiri dari 2 PID controller masing masing primer dan sekunder serta persamaan 2 buah *plant* yakni Sistem pengaturan flow dan system pengaturan level seperti pada gambar berikut:

Untuk mengatur *PID controller* dengan menggunakan kontrol kaskade, pertama-tama kita perlu mengidentifikasi sistem yang akan dikendalikan dan menentukan apa yang akan menjadi variabel output dan input. Kemudian, kita perlu menentukan kontroler primer dan kontroler sekunder, serta mengatur parameter PID untuk setiap kontrolir.

Setelah kontrolir telah dikonfigurasi, langkah selanjutnya adalah mengukur respon sistem terhadap kontroler dan mengoptimalkan parameter PID agar sistem dapat beroperasi dengan stabil dan memberikan hasil yang diinginkan. Ini bisa dilakukan dengan menggunakan teknik seperti trial and error, atau dengan menggunakan metode matematik yang lebih rumit seperti Ziegler-Nichols atau Cohen-Coon.

Setelah parameter PID telah dioptimalkan, kita dapat memantau sistem secara terus-menerus untuk memastikan bahwa sistem tetap stabil dan memberikan hasil yang diinginkan. Jika diperlukan, kita dapat mengubah parameter PID secara berkala untuk menyesuaikan diri dengan perubahan kondisi sistem atau tujuan kontrol.

Dengan menggunakan kontrol kaskade, kita dapat meningkatkan keandalan dan stabilitas sistem dengan mengendalikan beberapa variabel *output* secara bersamaan. Ini memungkinkan kita untuk mencapai hasil yang lebih baik daripada menggunakan satu PID controller saja.

### Tuning PID dengan Metode Particle Swarm Optimization

Tuning PID menggunakan PSO merupakan salah satu metode yang dapat digunakan untuk mencari nilai konstanta PID (Proportional, Integral, dan Derivative) yang optimal untuk mengendalikan sistem dinamis. Metode ini memanfaatkan kinerja dari kontroler PID yang sudah ada dengan mengoptimalkan nilai konstanta PID untuk mencapai performa yang lebih baik.

Dalam penerapannya, PSO akan mengoptimalkan nilai konstanta PID dengan memperhitungkan nilai error sistem, waktu respon, dan overshoot. PSO akan menggunakan partikel-partikel virtual yang mewakili nilai konstanta PID, dan mencari solusi terbaik dengan mengikuti aturan-aturan yang sama seperti kelompok serangga dalam lingkungannya. Nilai konstanta PID yang dihasilkan dari proses optimisasi menggunakan PSO diharapkan dapat meningkatkan performa kontroler PID dalam mengendalikan sistem dinamis.

Diagram

Description automatically generated

Berikut ini adalah algoritma tuning PID menggunakan PSO dalam bentuk pseudo code yang terdapat input, output, variabel, dan parameter:

**ALGORITME 3. Algoritme Tuning PID menggunakan PSO**

INPUT:

* Transfer function (TF)
* Jumlah partikel (N)
* Nilai w (inertia weight)
* Nilai c1 (cognitive weight)
* Nilai c2 (social weight)
* Nilai error konvergensi (error\_threshold)
* Batas maksimum iterasi (max\_iter)

OUTPUT:

* Nilai PID yang teroptimal

VARIABLE:

* velocity[N]
* position[N]
* error[N]
* fitness[N]
* best\_point[N]
* best\_fitness[N]

PARAMETER:

* Nilai kecepatan awal (initial\_velocity)
* Nilai titik awal (initial\_point)

ALGORITMA:

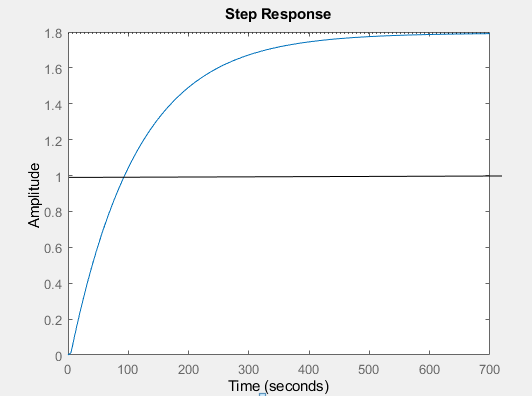
1. Set initial\_velocity dan initial\_point sebagai nilai awal untuk setiap partikel
2. Untuk setiap partikel:
   * Hitung error menggunakan TF dan nilai PID pada posisi partikel tersebut
   * Hitung fitness menggunakan nilai error
   * Set best\_point sebagai posisi partikel yang memiliki fitness terbaik
   * Set best\_fitness sebagai fitness terbaik yang diperoleh
3. Ulangi langkah 2 hingga kriteria konvergensi tercapai atau iterasi mencapai batas maksimum
4. Set nilai PID yang teroptimal sebagai posisi partikel yang memiliki fitness terbaik (best\_fitness)
5. Selesai.Top of Form

# Hasil Penelitian dan Pembahasan

## Hasil Pemodelan dari Sistem Pengaturan Level

Berdasarkan hasil pemodelan sistem pada Bab 3 didapatkan model akhir plant seperti pada Persamaan 4.1

Dilakukan pengujian sistem umpan balik tertutup menggunakan sinyal unit step pada Persamaan 4.1 didapatkan hasil respon pada Gambar 4.1.



Berdasarkan respon loop terbuka yang diperoleh dari pemodelan matetasi sistem, masih terdapat error steady state yang besar, yaitu 0.75. Dengan adanya permasalah tersebut, perlu digunakan kontroler untuk meminimalkan kesalahan waktu tunak pada sistem.

Pada sistem tersebut juga diperlukan kontroler yang dapat mengurangi efek pembebaban yang berasal dari perubahan nilai beban R. Perubahan nilai tersebut disebabkan oleh perubahan bukaan control valve pada sistem.

## Hasil Pemodelan dari Sistem dari Sistem Pengaturan Flow

Berdasarkan hasil pemodelan sistem pada Bab 3 didapatkan model akhir plant seperti pada Persamaan 4.2

Dilakukan pengujian *open loop* menggunakan sinyal unit step pada Persamaan 4.2 didapatkan hasil respon pada Gambar 4.1. Gambar 4.1 Respon open loop sistem pengaturan flow tanpa kontroler.

Chart

Description automatically generated

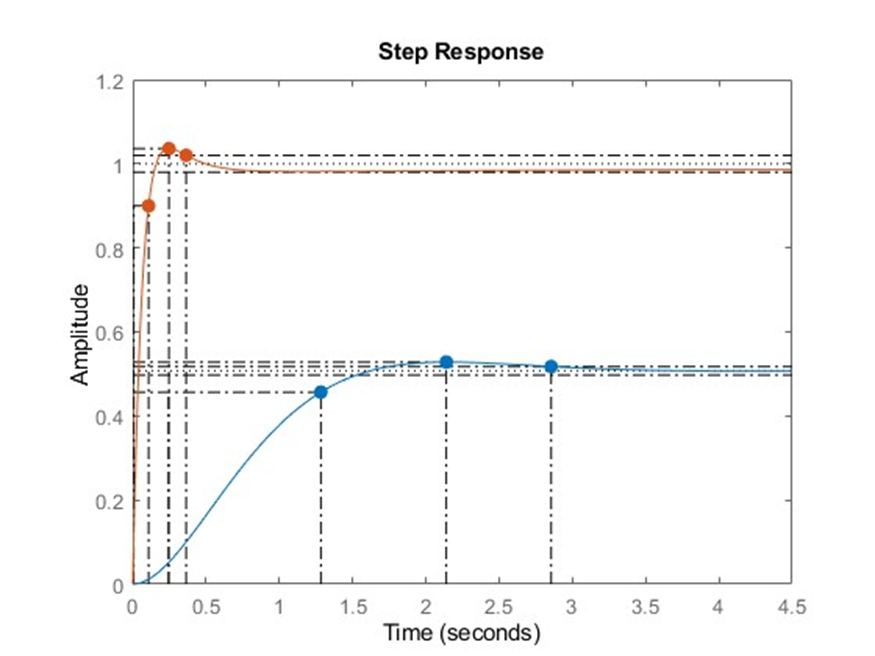
Diperoleh hasil karakteristik system tersebut adalah :

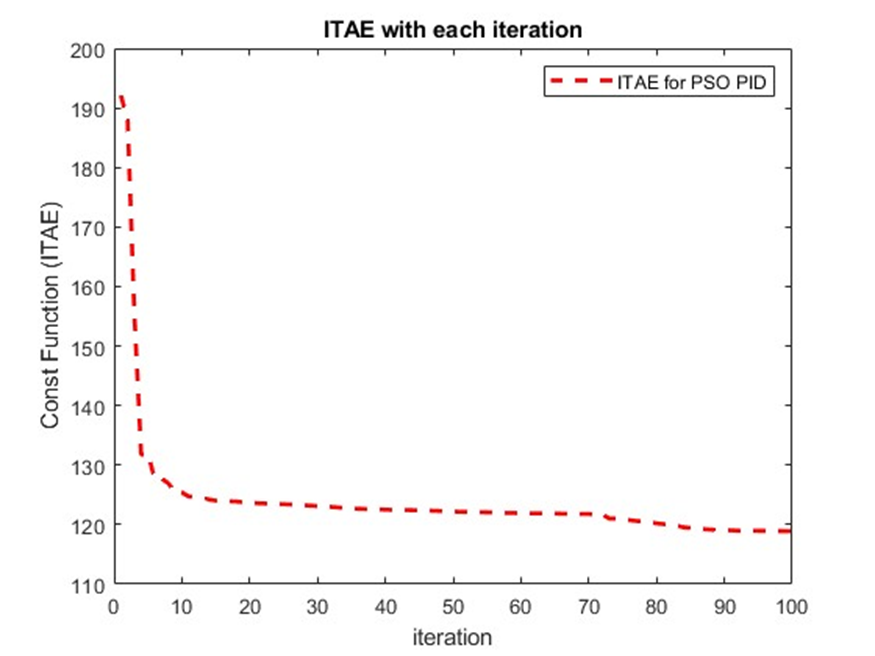
|  |  |
| --- | --- |
| RiseTime | 2.3516 |
| TransientTime | 4.1141 |
| SettlingTime | 4.1141 |
| SettlingMin | 0.8728 |
| SettlingMax | 0.9652 |
| Overshoot | 0 |
| Undershoot | 0 |
| Peak: | 0.9652 |
| PeakTime: | 6.7395 |

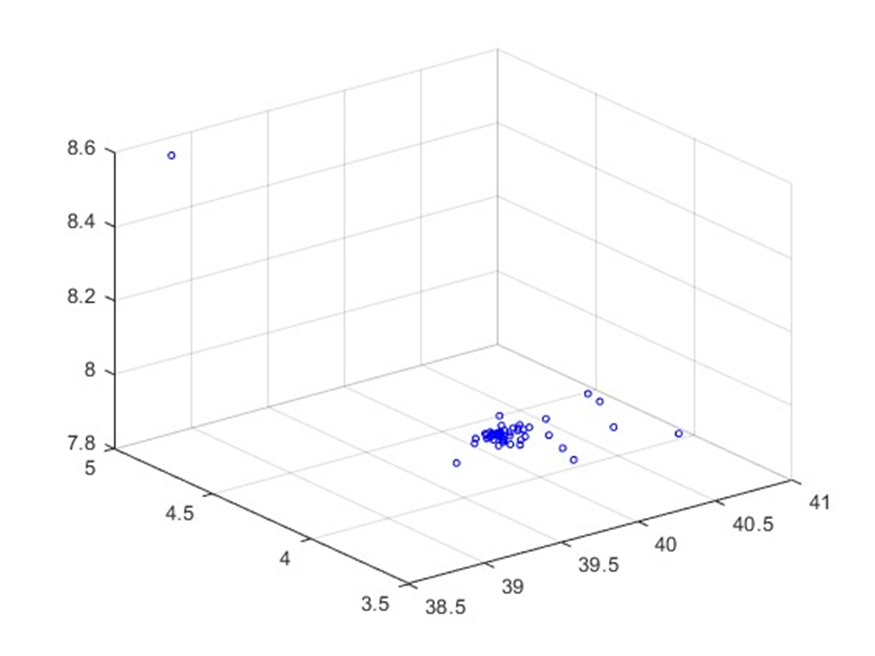
Dari data tersebut steady state pada 0.964 dari input set point atau sinyal step 1 sehingga didapat error steady state sebesar 1 – 0.964 = 0.036. Kesalahan keadaan tunak tersebut yang menjadikan dasar perlunya kontroler pada sistem pengaturan level PCT-100. Spesifikasi yang diinginkan adalah mampu menghilangkan kesalahan keadaan tunak serta mampu mempercepat respons sistem.

## Hasil Pengujian Tuning PID dengan PSO Inner Loop

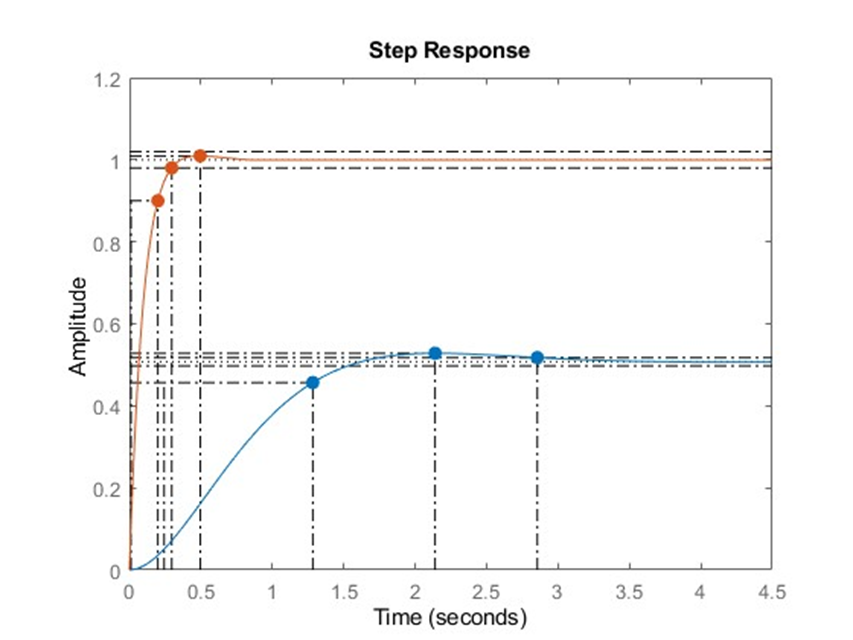
### Pengujian Variasi w=0.4.



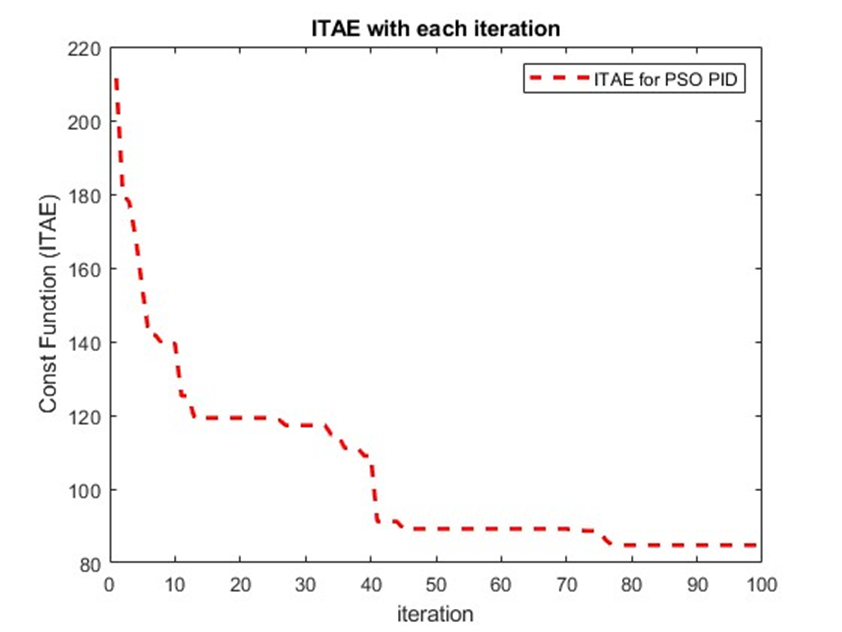




### Pengujian Variasi w=0.8



Chart, scatter chart

Description automatically generated

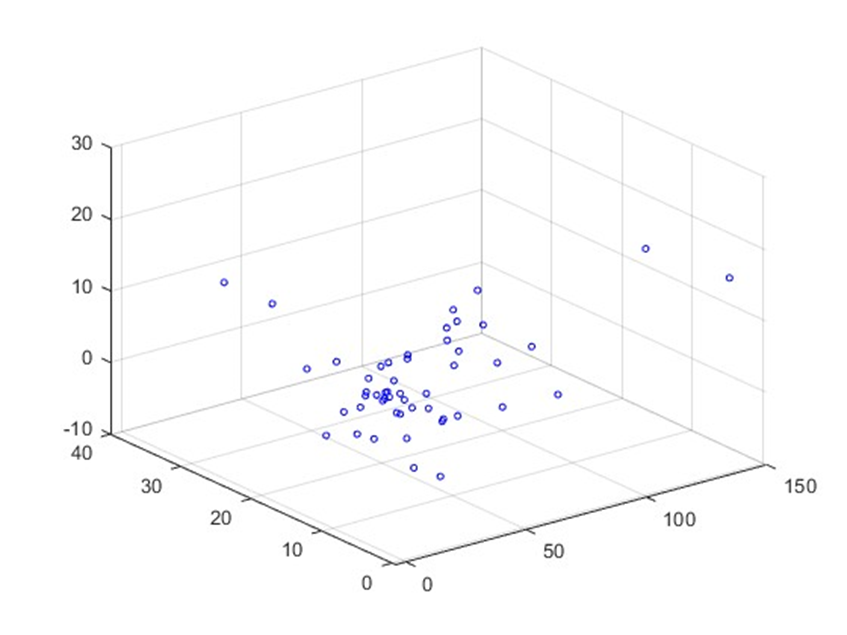
### Pengujian Variasi w = 0.9

Chart

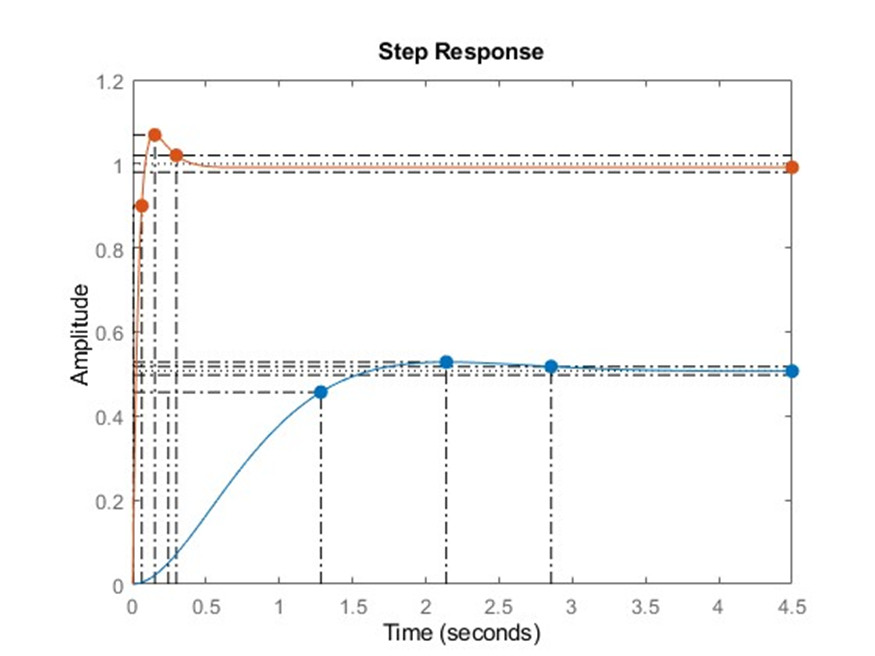
Description automatically generated

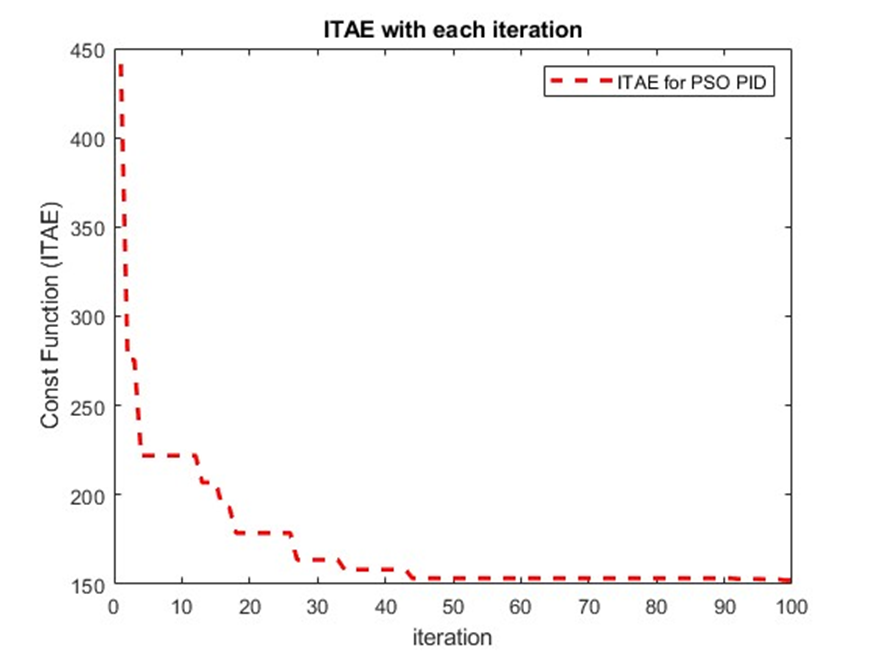
Chart, line chart

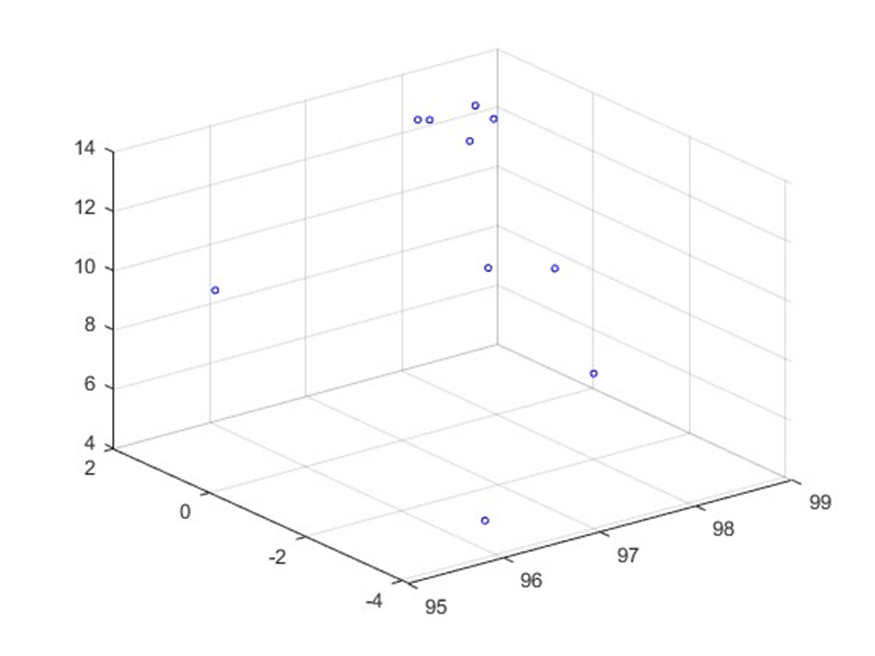
Description automatically generated



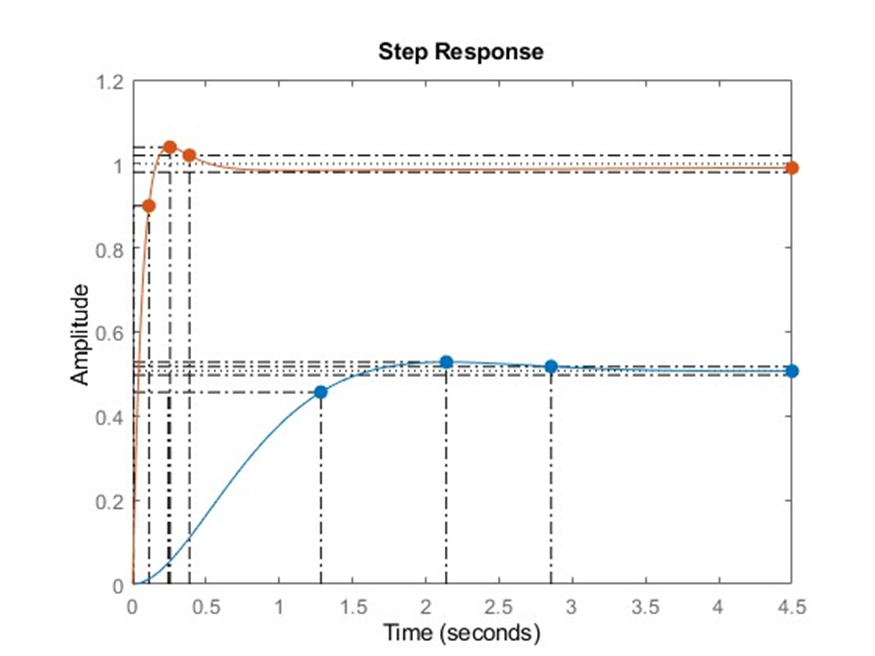
### Pengujian Jumlah Partikel (N) = 10

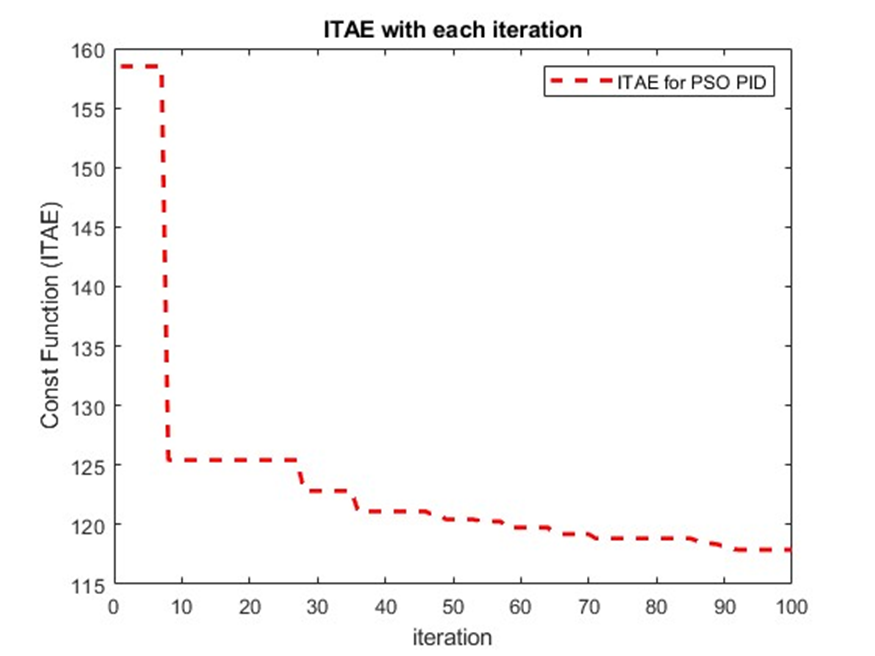


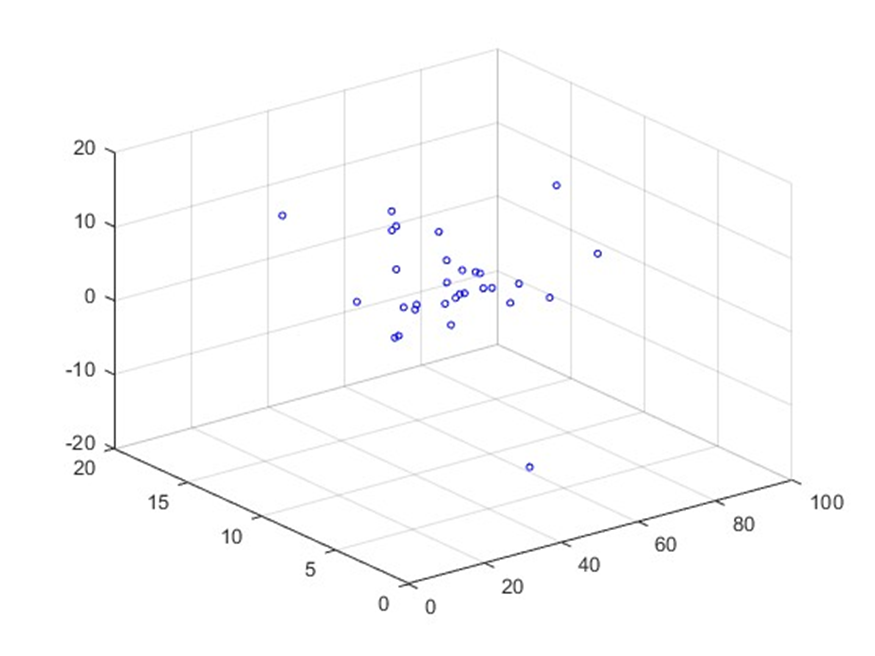




### Pengujian Jumlah Partikel (N)=30







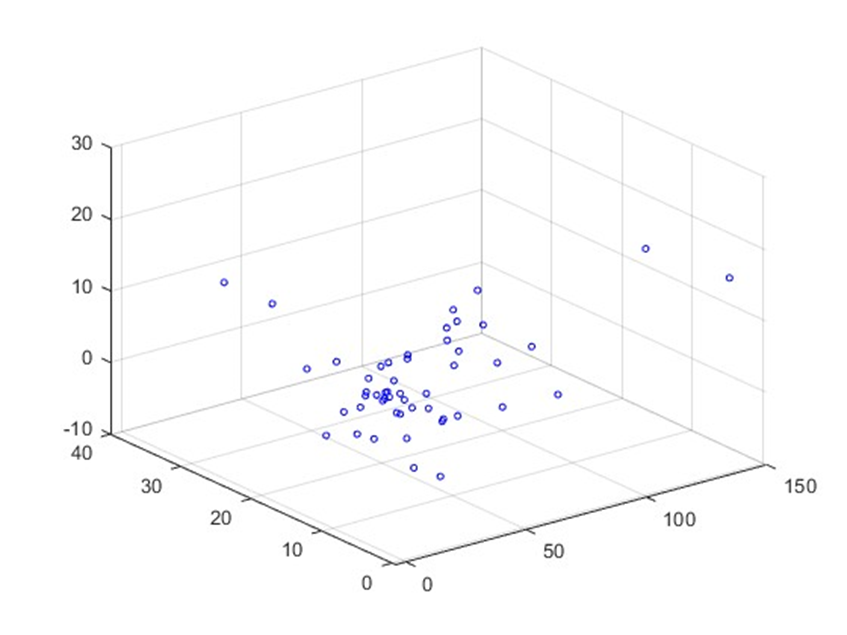
### Pengujian Jumlah Partikel (N)=50

Chart

Description automatically generated

Chart, line chart

Description automatically generated



## Hasil Pengujian Tuning PID dengan PSO pada Outer Loop

## Implementasi Simulasi Tuning PID dengan PSO pada Sistem Pengaturan Cascade Level dan Flow

## Pembahasan

# Kesimpulan dan Saran

## Kesimpulan

Kesimpulan yang dapat diambil dalam tugas akhir ini antara lain, adalah :

* 1. Parameter PID hasil penalaan dengan algoritma PSO adalah Kp=7,60, Ki=0,66, Kd = 0,56
  2. Fitness value yang paling optimal sebesar 39.9901 pada system pengaturan flow dan 41.0909 pada system pengaturan level

## Saran

Adapun saran yang diberikan pada tugas akhir ini adalah sebagai berikut:

* 1. Pada plant yang digunakan yakni Sistem pengaturan Flow Plant PCT 100 seharusnya menggunakan valve flow control sehingga bisa menghasilkan respons yang lebih baik adanya overshoot pada system ini dikarenakan flow dikontrol oleh pompa listrik sehingga waktu starting mengalami overshoot akibat dari putaran pompa tersebut

DAFTAR PUSTAKA

Ang, K. H., Chong, G., & Li, Y. (2005). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, *13*(4), 559–576. https://doi.org/10.1109/TCST.2005.847331

Bytronic Limited. (n.d.). *"P r o c e s s C o n t r o l T e c h n o l o g y P C T - 1 0 0 " Manual Book*. Bytronic Limited.

Guong, R. L. L., Hisham, S. B., Elamvazuthi, I., Aole, S., Parasuraman, S., & Khan, M. K. A. A. (2018). PID Tuning of Process Plant using Particle Swarm Optimization. *2018 IEEE 4th International Symposium in Robotics and Manufacturing Automation, ROMA 2018*, *1*(1), 0–5. https://doi.org/10.1109/ROMA46407.2018.8986712

Ibrahim, H. E. A., Hassan, F. N., & Shomer, A. O. (2014). Optimal PID control of a brushless DC motor using PSO and BF techniques. *Ain Shams Engineering Journal*, *5*(2), 391–398. https://doi.org/10.1016/j.asej.2013.09.013

Jakoubek, P. (2009). Experimental Identification of Stabile Nonoscillatory Systems from Step-Responses by Selected Methods. *Student Conderence STC 2009*. http://stc.fs.cvut.cz/history/2009/sbornik/Papers/pdf/JakoubekPavel-304494.pdf

Kawasan, G., Serpong, P., & Tlp, T. (2011). *Penerapan pemodelan dan metode kurva reaksi proses untuk mengidentifikasi sistem duress*. 230–236.

Kumar, C. A., Nair, N. K., & Engineering, E. (2010). Multi-Objective Pid Controller Based on Adaptive Weighted Pso With Application To Steam Temperature Control in Boilers. *International Journal of Engineering Science and Technology*, *2*(7), 3179–3184.

Ogata, K. (2010). Modern Control Engineering 5th Edition. In *Prentice Hall PTR* (5th ed.).

Pano, V., & Ouyang, P. R. (2014). PSO gain tuning for position domain PID controller. *4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2014*, 377–382. https://doi.org/10.1109/CYBER.2014.6917493

Pillay, N. (2013). a Particle Swarm Optimization Approach for Tuning of Siso Pid Control Loops. *Engineering Research on Soft Computing Methods*, 235–258.

Pritandi, D. A. P. A. (2016). Perancangan Kontroler PID-Fuzzy untuk Sistem Pengaturan Cascade Level dan Flow pada Basic Process Rig 38-100. *Jurnal Teknik ITS*, *5*(2). https://doi.org/10.12962/j23373539.v5i2.16179

Tridianto, E., Ariwibowo, T. H., Almasa, S. K., & Prasetya, H. E. G. (2017). Cascaded PID temperature controller for FOPDT model of shell-and-tube heat exchanger based on Matlab/Simulink. *Proceedings IES-ETA 2017 - International Electronics Symposium on Engineering Technology and Applications*, *2017*-*Decem*, 185–191. https://doi.org/10.1109/ELECSYM.2017.8240400

Vilanova, R., & Arrieta, O. (2008). PID TUNING FOR CASCADE CONTROL SYSTEM DESIGN ETSE , Universitat Aut `. *Canadian Conference on Electrical and Computer Engineering, 2008. CCECE 2008.*, *1*, 1775–1778.

Wen, S., Deng, M., bi, S., & Wang, D. (2012). Operator-based robust nonlinear control and its realization for a multi-tank process by using a distributed control system. *Transactions of the Institute of Measurement and Control*, *34*(7), 891–902. https://doi.org/10.1177/0142331211424427

Xiang, Z., Ji, D., Zhang, H., Wu, H., & Li, Y. (2019). A simple PID-based strategy for particle swarm optimization algorithm. *Information Sciences*, *502*, 558–574. https://doi.org/10.1016/j.ins.2019.06.042

Xiong, A., & Fan, Y. (2007). Application of a PID controller using MRAC techniques for control of the DC electromotor drive. *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007*, 2616–2621. https://doi.org/10.1109/ICMA.2007.4303969

LAMPIRAN

**Lampiran Tuning PID Inner Loop Menggunakan Algoritma PSO**

clc; clear all; close all;

num = [1.03];

den = [0.468 1.39 1];

G = tf(num,den);

Gf = feedback(G,1);

step(Gf)

hold on

%PSO Constant Parameters

c1=1.5; c2=1; w=0.9; particles=100; iteration=100; Var=3;

% Search Space

a=0; %lower bound

b=100; %upper bound

% Optimization Steps

c\_cf=0;

%initialization

for m=1:particles

for n=1:Var

v(m,n)=0; %Velocity Particles

x(m,n)=a+rand\*(b-a); %Position Particles

xp(m,n)= x(m,n);

end

%Model Parameters

kp = x(m,1);

ki = x(m,2);

kd = x(m,3);

%Simulation Model

Gc=pid(kp,ki,kd);

Gcf=feedback(Gc\*G,1);

y=step(Gcf);

% ITAE (Objective Function)

ff1=0;

sim1=size(y);

for m1= 1:sim1

ff1=ff1+((abs(y(m1)-1))\*m1);

end

ITAE(m)=ff1;

end

%Find The Best Value

[Best\_Performance, location]=min(ITAE);

fg=Best\_Performance

xg(1)=x(location,1);

xg(2)=x(location,2);

xg(3)=x(location,3);

for i=1:iteration % No Of Repetition

for m=1:particles % No Of Particles

for n=1:Var % No Of Var

v(m,n)=(w\*v(m,n))+(c1\*rand\*(xp(m,n)-x(m,n)))+(c2\*rand\*(xg(n)-x(m,n))); % Update Velocity

x(m,n)=x(m,n)+v(m,n); % Update Position

end

positions(m,1) = x(m,1); % position of kp for particle m

positions(m,2) = x(m,2); % position of ki for particle m

positions(m,3) = x(m,3); % position of kd for particle m

%Check Bound

for n=1:Var

if x(m,n)<a

x(m,n)=a;

end

if x(m,n)>b

x(m,n)=b;

end

end

% Model Parameters

kp=x(m,1);

ki=x(m,2);

kd=x(m,3);

% Simulation Model

Gc=pid(kp,ki,kd);

Gcf=feedback(Gc\*G,1);

y=step(Gcf);

% ITAE (Objective Function)

ff1=0;

sim1=size(y);

for m1=1:sim1

ff1=ff1+((abs(y(m1)-1))\*m1);

end

ITAEp(m)=ff1;

% Compare Local

if ITAEp(m)<ITAE(m)

ITAE(m)=ITAEp(m);

xp(m,1)=x(m,1);

xp(m,2)=x(m,2);

xp(m,3)=x(m,3);

end

end

[B\_fg,location]=min(ITAE);

% Compare Global

if B\_fg<fg

fg=B\_fg; % New Global Value

xg(1)=xp(location,1); % Position of Var 1

xg(2)=xp(location,2); % Position of Var 2

xg(3)=xp(location,3); % Position of Var 3

end

c\_cf = c\_cf+1;

best\_cf\_ac(c\_cf)=fg;

end

Min\_ITAE=fg

kp=xg(1)

ki=xg(2)

kd=xg(3)

Gc=pid(kp,ki,kd);

Gcf=feedback(Gc\*G,1);

step(Gcf);

t\_cf=1:c\_cf;

figure

plot(t\_cf,best\_cf\_ac,'r--','LineWidth',2), xlabel('iteration'), ylabel('Const Function (ITAE)')

legend('ITAE for PSO PID')

title('ITAE with each iteration')

figure

scatter3(positions(:,1), positions(:,2), positions(:,3));

xlabel('kp');

ylabel('ki');

zlabel('kd');

scatter3(positions(:,1), positions(:,2), positions(:,3), 10, 'blue');

% Plot step response and calculate steady-state error

step\_info = stepinfo(Gcf); % Get step response information

y\_inf = step\_info.Peak; % Get final value of step response

t\_inf = step\_info.RiseTime; % Get time corresponding to final value of step response

y\_inf = step\_info.SettlingMax; % Get final value of step response

t\_inf = step\_info.SettlingTime; % Get time corresponding to final value of step response

ess = abs(1 - y\_inf); % Calculate steady-state error as absolute difference between desired value (1) and final value of step response

% Calculate RMSE for a sequence of steady-state errors

rmse = sqrt(ess^2); % Calculate RMSE as the square root of the mean of the squares of the steady-state errors

**Lampiran Tuning PID Outer Loop Menggunakan Algoritma PSO**

clc; clear all; close all;

num = [10.9 38.52 18.49];

den = [51.44 820.9 2477 1155 10.3];

G=tf(num,den);

Gf = feedback(G,1);

step(Gf)

hold on

%PSO Constant Parameters

c1=1.5; c2=1; w=0.9; particles=50; iteration=100; Var=3;

% Search Space

a=0; %lower bound

b=100; %upper bound

% Optimization Steps

c\_cf=0;

%initialization

for m=1:particles

for n=1:Var

v(m,n)=0; %Velocity Particles

x(m,n)=a+rand\*(b-a); %Position Particles

xp(m,n)= x(m,n);

end

%Model Parameters

kp = x(m,1);

ki = x(m,2);

kd = x(m,3);

%Simulation Model

Gc=pid(kp,ki,kd);

Gcf=feedback(Gc\*G,1);

y=step(Gcf);

% ITAE (Objective Function)

ff1=0;

sim1=size(y);

for m1= 1:sim1

ff1=ff1+((abs(y(m1)-1))\*m1);

end

ITAE(m)=ff1;

end

%Find The Best Value

[Best\_Performance, location]=min(ITAE);

fg=Best\_Performance

xg(1)=x(location,1);

xg(2)=x(location,2);

xg(3)=x(location,3);

for i=1:iteration % No Of Repetition

for m=1:particles % No Of Particles

for n=1:Var % No Of Var

v(m,n)=(w\*v(m,n))+(c1\*rand\*(xp(m,n)-x(m,n)))+(c2\*rand\*(xg(n)-x(m,n))); % Update Velocity

x(m,n)=x(m,n)+v(m,n); % Update Position

end

positions(m,1) = x(m,1); % position of kp for particle m

positions(m,2) = x(m,2); % position of ki for particle m

positions(m,3) = x(m,3); % position of kd for particle m

%Check Bound

for n=1:Var

if x(m,n)<a

x(m,n)=a;

end

if x(m,n)>b

x(m,n)=b;

end

end

% Model Parameters

kp=x(m,1);

ki=x(m,2);

kd=x(m,3);

% Simulation Model

Gc=pid(kp,ki,kd);

Gcf=feedback(Gc\*G,1);

y=step(Gcf);

% ITAE (Objective Function)

ff1=0;

sim1=size(y);

for m1=1:sim1

ff1=ff1+((abs(y(m1)-1))\*m1);

end

ITAEp(m)=ff1;

% Compare Local

if ITAEp(m)<ITAE(m)

ITAE(m)=ITAEp(m);

xp(m,1)=x(m,1);

xp(m,2)=x(m,2);

xp(m,3)=x(m,3);

end

end

[B\_fg,location]=min(ITAE);

% Compare Global

if B\_fg<fg

fg=B\_fg; % New Global Value

xg(1)=xp(location,1); % Position of Var 1

xg(2)=xp(location,2); % Position of Var 2

xg(3)=xp(location,3); % Position of Var 3

end

c\_cf = c\_cf+1;

best\_cf\_ac(c\_cf)=fg;

end

Min\_ITAE=fg

kp=xg(1)

ki=xg(2)

kd=xg(3)

Gc=pid(kp,ki,kd);

Gcf=feedback(Gc\*G,1);

step(Gcf);

t\_cf=1:c\_cf;

figure

plot(t\_cf,best\_cf\_ac,'r--','LineWidth',2), xlabel('iteration'), ylabel('Const Function (ITAE)')

legend('ITAE for PSO PID')

title('ITAE with each iteration')

figure

scatter3(positions(:,1), positions(:,2), positions(:,3));

xlabel('kp');

ylabel('ki');

zlabel('kd');

scatter3(positions(:,1), positions(:,2), positions(:,3), 10, 'blue');

% Plot step response and calculate steady-state error

step\_info = stepinfo(Gcf); % Get step response information

y\_inf = step\_info.Peak; % Get final value of step response

t\_inf = step\_info.RiseTime; % Get time corresponding to final value of step response

y\_inf = step\_info.SettlingMax; % Get final value of step response

t\_inf = step\_info.SettlingTime; % Get time corresponding to final value of step response

ess = abs(1 - y\_inf); % Calculate steady-state error as absolute difference between desired value (1) and final value of step response

% Calculate RMSE for a sequence of steady-state errors

rmse = sqrt(ess^2); % Calculate RMSE as the square root of the mean of the squares of the steady-state errors

BIODATA PENULIS



Penulis dilahirkan di Wonogiri, 1 Januari 2000, merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Mambaul Hikmah Selogiri, SDIT Al Huda Wonogiri, SMPN 1 Sukoharjo dan SMAN 1 Sukoharjo. Setelah lulus dari SMAN 1 Sukoharjo tahun 2018, Penulis mengikuti SNMPTN dan diterima di Departemen Teknik Elektro FTEIC - ITS pada tahun 2018 dan terdaftar dengan NRP 07111840000029.

Di Departemen Teknik Elektro Penulis sempat aktif di beberapa kegiatan Organisasi yang diselenggarakan oleh ITS yakni Staff JMMI pada 2018, Menjabat Sekretaris Lembaga Dakwah Islam KALAM ELITS dan aktif sebagai Asisten Laboratorium Kontrol dan Otomasi. Selain itu penulis juga aktif mengikuti PKM.