

**PEMODELAN SISTEM PENGENDALI PID DENGAN  
METODE CIANCONE BERBASIS MATLAB SIMULINK  
PADA SISTEM *PRESSURE PROCESS RIG 38-714***

***SYSTEM MODELLING WITH PID CONTROLLER APPLYING  
CIANCONE METHOD USING MATLAB SIMULINK ON  
PRESSURE PROCESS RIG 38-714 SYSTEM***

**Ikhwannul Kholis**

**Universitas Mpu Tantular  
Ikkholis27@gmail.com**

**Abstrak**

Pengendalian sistem di dunia industri sangat diperlukan dalam keadaan *offline* sehingga proses produksi tidak terganggu. Pemodelan sistem bertujuan untuk melakukan simulasi sistem dan mendesain pengendali untuk memperbaiki respon sistem, salah satunya dengan menggunakan pengendali PID. Pemodelan sistem dan Pengendali PID dapat dilakukan dengan metode Ciancone. Salah satu sistem yang sering digunakan untuk pemodelan adalah sistem *Pressure Process Rig*. Pada penelitian ini, dilakukan pemodelan sistem dan desain pengendali PID dengan metode Ciancone pada sistem *Pressure Process Rig 38-714* dengan menggunakan Matlab Simulink.

**Kata Kunci:** *pengendali PID, pemodelan sistem, pressure process rig 38-714, metode Ciancone.*

**Abstract**

*Control System in industry is indispensable in offline mode to keep the production process uninterrupted. Modelling System aims to simulate the systems and to design controller to improve the response system, such as PID Controller. Modelling System and PID Controller can be constructed by applying Ciancone method. The frequently used system for modeling is Pressure Process Rig System. In this study, Modelling System and Designing PID Controller were conducted by applying Ciancone method in Pressure Process Rig 38-714 System using Matlab Simulink.*

**Keywords:** *PID Controller, System Modelling, Pressure Process Rig 38-714, Ciancone Method*

**Tanggal Terima Naskah : 01 Juni 2016**  
**Tanggal Persetujuan Naskah : 10 Januari 2017**

**1. PENDAHULUAN**

Pengukuran dan kendali proses merupakan hal yang sangat penting dalam proses industri. Pengukuran pada proses industri dilakukan untuk memperoleh *variable* yang dikontrol sehingga keluaran proses dapat dimanipulasi sesuai dengan keluaran yang

diinginkan. Kendali proses merupakan suatu teknik kendali pada proses industri. Suatu proses industri biasanya memiliki *plant* yang tidak dapat diketahui secara langsung proses di dalamnya. Sementara itu, keluaran dari sistem perlu dikendalikan untuk mendapatkan keluaran yang diharapkan [1]. Hal ini yang menjadi peran pengukuran dan pengendalian pada proses industri. Pengukuran diperlukan untuk mengukur *variable* yang dikontrol sebagai *feedback* kepada sistem. Pengendalian proses berperan dalam memanipulasi keluaran sistem sehingga keluaran sistem sesuai dengan keluaran yang diharapkan [2].

Pengendalian proses sangat erat kaitannya dengan proses identifikasi. Proses identifikasi merupakan pengenalan suatu sistem atau *plant* yang akan dikendalikan. Hal ini dilakukan untuk membuat kontroler secara *offline* sehingga proses industri tidak terganggu [3]. Simulasi dilakukan dengan mengidentifikasi sistem menjadi suatu *transfer function* sesuai dengan *plant*. Oleh karena itu, proses identifikasi merupakan salah satu faktor penting yang menunjang keberhasilan dalam merencanakan suatu sistem kendali yang stabil, *robust*, dan mampu beradaptasi dengan baik terhadap lingkungan [4].

Desain kontroler bertujuan untuk memanipulasi *input* sehingga menghasilkan keluaran yang sesuai dengan keluaran yang diharapkan. Hal ini dapat dilakukan dengan membuat *transfer function* yang sesuai dengan *plant* dengan menggunakan data yang dihasilkan oleh *plant*. Dengan ilmu pengendalian proses, sistem dapat diidentifikasi menjadi *transfer function*. Selanjutnya, desain kontroler dilakukan untuk menghasilkan keluaran yang diharapkan.

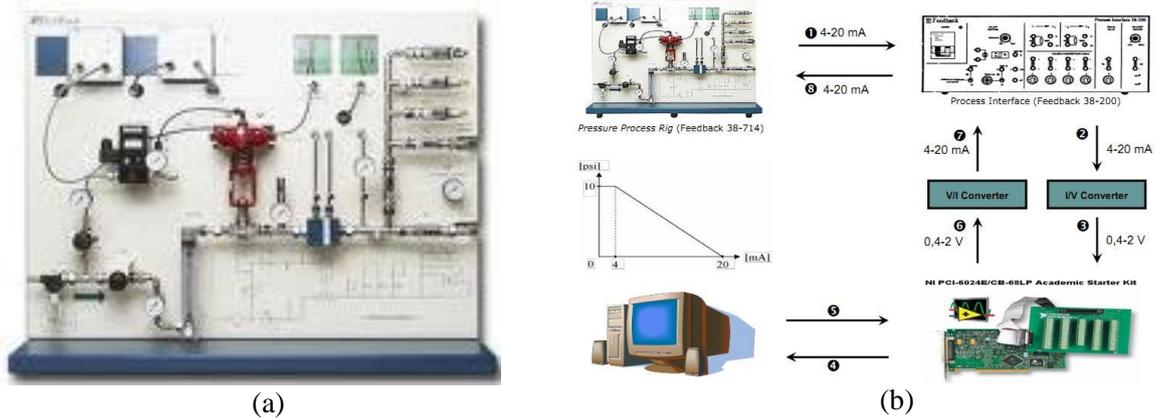
Proses identifikasi sistem dapat dilakukan dengan beberapa cara, salah satunya adalah *Process Reaction Curve* (PRC). PRC merupakan metode untuk mencari *transfer function* suatu sistem dengan menggunakan informasi dari grafik *Response* sistem. Dengan demikian, sistem dapat diidentifikasi untuk dapat dilakukan desain pengendali agar menghasilkan keluaran yang diharapkan [5].

## 2. KONSEP DASAR

### 2.1 *Pressure Process Rig*

*Pressure Process Rig* 38-714 adalah peralatan produksi PROCON yang digunakan untuk mengenalkan dan mendemonstrasikan prinsip-prinsip proses pengukuran dan pengendalian [6]. Sistem yang disediakan adalah sistem kendali *pneumatic* sebagaimana yang umumnya digunakan dalam industri.

Secara umum *Pressure Process Rig* 38-714 terdiri dari jalur-jalur pipa yang terhubung pada *Pneumatic Control Valve*, *Orifice Block*, *Flowmeter*, *pressure tapping*, *regulator*, indikator tekanan, dan *valve*. Regulator R1 digunakan untuk mengendalikan tekanan yang diukur oleh G1. Regulator R2 digunakan untuk mengatur tekanan yang diukur oleh G3, G4 atau G5. Indikator tekanan G6 digunakan untuk menunjukkan tekanan pada *air receiver*. *Output* yang dikendalikan pada penelitian ini adalah *flow* dari *Pressure Process Rig*.



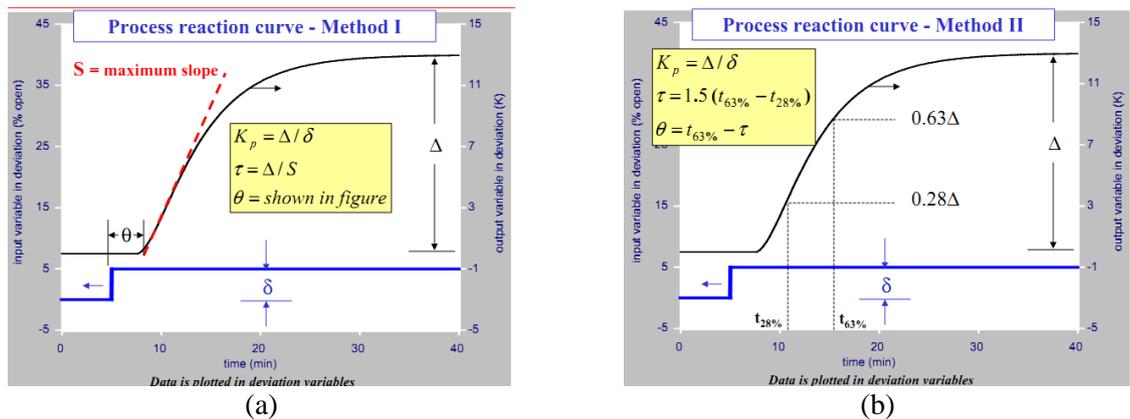
Gambar 1. (a) *Pressure Process Rig*; (b) Cara kerja *Pressure Process Rig*

Pada Gambar 1.b, terlihat cara kerja *Pressure Process Rig*. Pada Gambar 1.b., terlihat komputer memberikan sinyal digital yang masuk ke DAC, selanjutnya dikonversi menjadi tegangan analog antara 0.4-2V, masuk ke rangkaian V/I sehingga dikonversi menjadi arus 4-20 mA. Arus ini akan masuk ke *plant (pressure process rig)* dan diubah dengan *I/P converter* untuk mengatur *Pneumatic Control Valve*. Aliran yang akan keluar di *sense* oleh *differential pressure sensor* yang bekerja dengan prinsip Bernaulli. Dengan *transducer*, nilai aliran akan dikonversi menjadi sinyal elektrik (arus). Selanjutnya, arus dikonversi dengan rangkaian I/V menjadi tegangan. Tegangan dikonversi oleh ADC menjadi sinyal digital agar dapat dibaca oleh komputer [7].

*Pressure Process Rig* digunakan untuk meneliti pengendalian proses, yaitu pengukuran *controlled variable (CV)*, membuat grafik keluaran, dan percobaan kontroler yang telah didesain. CV yang digunakan adalah tekanan dan kecepatan aliran udara. Pada akhirnya, penggunaan *Pressure Process Rig* dilakukan untuk mencoba kontroler yang didesain agar CV dapat mencapai nilai yang diharapkan dengan *error steady-state* yang kecil.

## 2.2 Process Reaction Curve

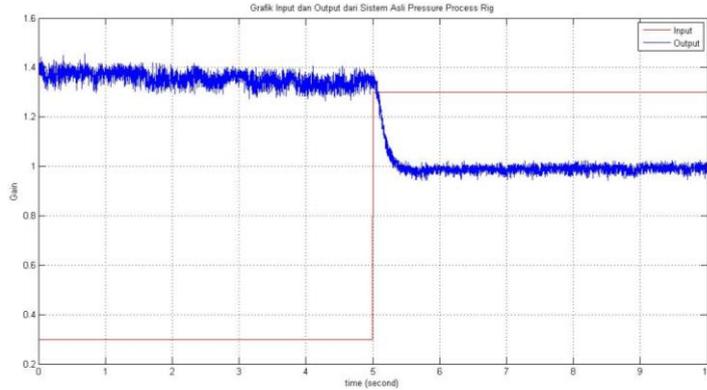
*Process Reaction Curve (PRC)* merupakan metode yang digunakan untuk mengidentifikasi suatu sistem dengan menggunakan grafik *Response* sistem tersebut. PRC yang digunakan pada penelitian ini berupa PRC orde satu [8]. Persamaan PRC orde satu adalah sebagai berikut.



Gambar 2. (a) *Process Reaction Curve – Method I*; (b) *Process Reaction Curve – Method II*

### 3. PEMBAHASAN

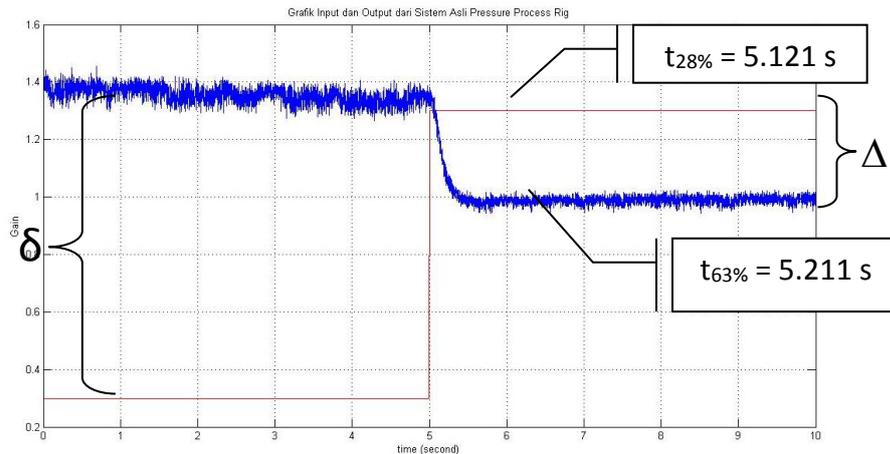
Pengambilan data pada *Pressure Process Rig* dilakukan untuk pemodelan sistem tersebut. Pengambilan data dilakukan dengan memberikan *input step* dengan *initial value* sebesar 0,3 dan *final value* sebesar 1 dengan perubahan *step* dilakukan pada detik ke-5. Data tersebut diolah menjadi grafik *response* sistem sebagai berikut.



Gambar 3. Respon sistem *Pressure Process Rig*

#### 3.1 Permodelan dengan Metode 1 (*Process Reaction Curve*)

Dari Gambar 3, dapat dilakukan pemodelan sistem sebagai berikut.



Gambar 4. Pemodelan sistem *Pressure Process Rig* dengan *Process Reaction Curve*

Dari Gambar 2, diberikan persamaan *Process Reaction Curve* untuk memperoleh *transfer function* pemodelan sistem *Pressure Process Rig*.

$$\Delta = FinalValue - InitialValue \dots\dots\dots [1]$$

$$Value_{63\%} = InitialValue + (\Delta \times 63\%) \dots\dots\dots [2]$$

$$Value_{28\%} = InitialValue + (\Delta \times 28\%) \dots\dots\dots [3]$$

$$\tau = 1.5(t_{63\%} - t_{28\%}) \dots\dots\dots [4]$$

$$\theta = t_{63\%} - \tau \dots\dots\dots [5]$$

$$Kp = \frac{\Delta}{\delta} \dots\dots\dots [6]$$

$$G(s) = \frac{Kp}{\tau s + 1} e^{-\theta s} \dots\dots\dots [7]$$

Dengan menggunakan Persamaan 1 sampai dengan persamaan 7, diperoleh data sebagai berikut.

Tabel 1. Data perhitungan *Process Reaction Curve* dari Grafik Gambar 4

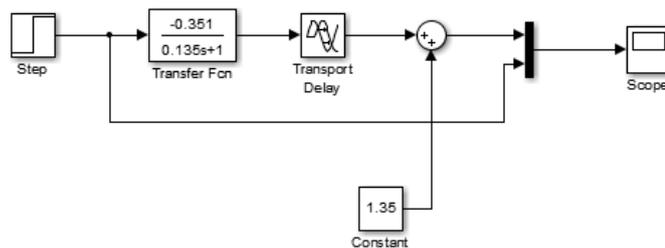
<i>Description</i>	<i>Value</i>	<i>Description</i>	<i>Value</i>
$\Delta$	-0,351	t(63%)	5,211
Value(63%)	1,129	t(28%)	5,121
Value(28%)	1,252	$\theta$	0,076
$\tau$	0,135	Kp	-0,351

Berdasarkan tabel 1, diperoleh *transfer function* berikut.

$$G(s) = \frac{Kp}{\tau s + 1} e^{-\theta s} \dots\dots\dots [8]$$

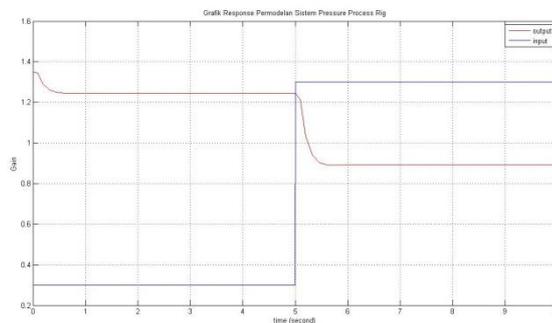
$$= \frac{-0.351}{0.135s + 1} e^{-0.076s}$$

Berdasarkan Gambar 4, data yang diperoleh memiliki data awal bernilai 1,35 sehingga digunakan *constant* untuk menaikkan sistem pada nilai *initial value system*, yaitu 1,35. Berdasarkan persamaan 8, dibuat diagram blok dengan menggunakan Simulink Matlab.



Gambar 5. Diagram blok pemodelan sistem

Pada diagram blok yang terlihat pada Gambar 5, digunakan *Transport Delay* untuk memodelkan *dead time* yang ada pada fungsi alih sistem sesuai dengan persamaan 8, yaitu  $e^{-0.076s}$ . Blok tersebut dijalankan di Matlab dengan memperoleh hasil berikut.

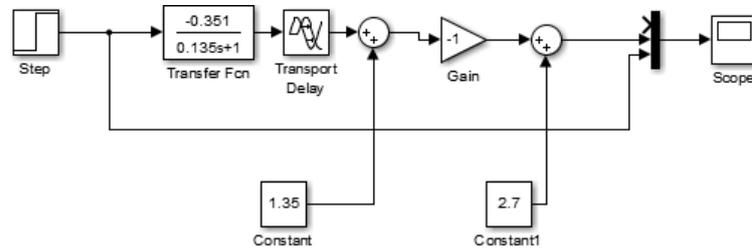


Gambar 6. Grafik pemodelan sistem dengan *Process Reaction Curve*

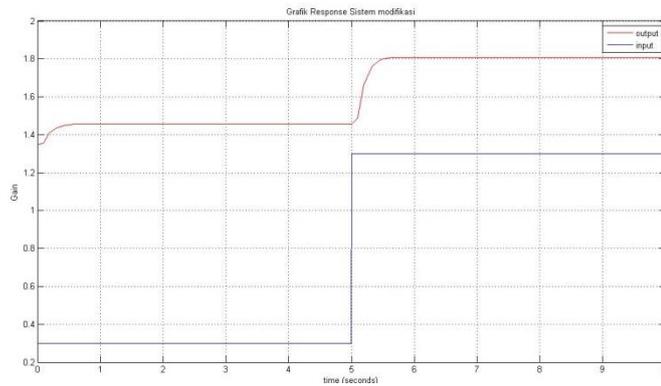
### 3.2 Desain Pengendali PID dengan Metode Ciancone

Untuk mendesain Pengendali PID dengan menggunakan metode Ciancone, diperlukan perubahan respon keluaran sistem. *Output* yang akan digunakan pada metode Ciancone adalah *output* yang mengikuti grafik *input* dengan menggunakan penambahan

konstanta yang memiliki nilai sebesar 2 kali konstanta awal. Hal ini dimaksudkan agar sistem memiliki *initial value* yang sama dengan data yang diperoleh dan sistem menghasilkan keluaran yang mengikuti grafik masukan. Berikut adalah gambar diagram blok sistem permodelan yang telah dimodifikasi.



Gambar 7. Diagram blok modifikasi untuk Metode Ciancone



Gambar 8. Grafik sistem pada Diagram Blok Gambar 7

Untuk mendesain pengendali Ciancone dengan parameter  $K_p$ ,  $\theta$ , dan  $\tau$ , diperlukan *transfer function* dari sistem pada Diagram Blok tersebut.

Tabel 2. Data perhitungan *Process Reaction Curve* dari Grafik Gambar 7

Description	Value	Description	Value
$\Delta$	0.351	t(63%)	5.211
Value(63%)	1.676	t(28%)	5.121
Value(28%)	1.553	$\theta$	0.076
$\tau$	0.135	$K_p$	0.351

sehingga diperoleh *transfer function system* adalah

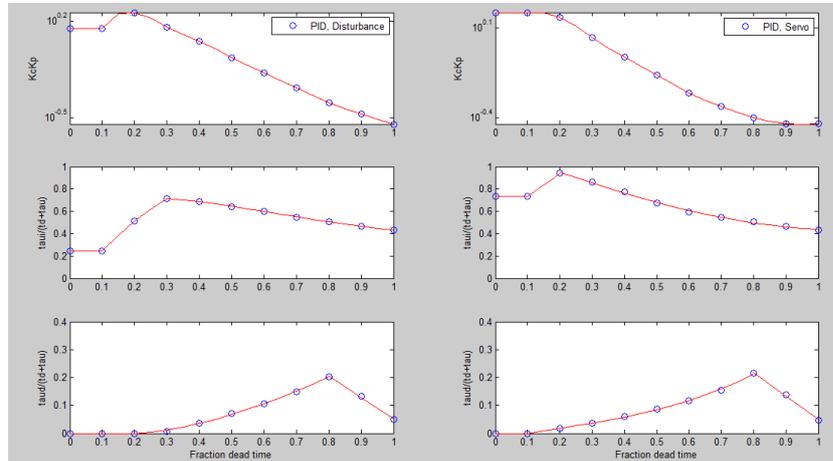
$$G(s) = \frac{K_p}{\tau s + 1} e^{-\theta s}$$

$$= \frac{0.351}{0.135s + 1} e^{-0.076s} \dots\dots\dots [9]$$

Selanjutnya, berdasarkan *transfer function*  $G(s)$  tersebut, diperoleh nilai  $K_p = 0,351$ ,  $\theta = 0,088$ , dan  $\tau = 0,132$ . Untuk mencari pengendali Ciancone, digunakan bentuk umum dari Pengendali Ciancone, yaitu

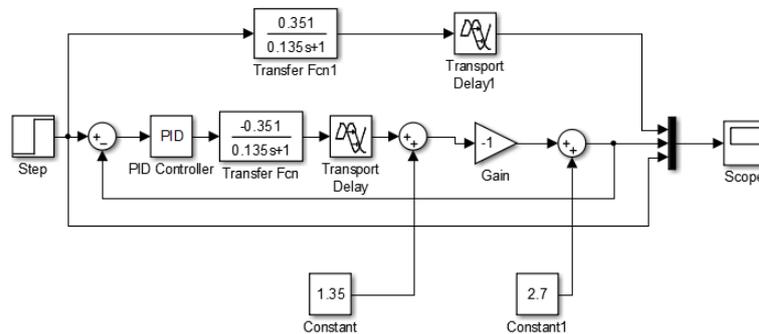
$$MV = K_c \left[ E(t) + \frac{1}{T_i} \int_0^{\infty} E(t)dt - T_d \frac{dCV}{dt} \right] + I \dots\dots\dots[10]$$

Untuk menentukan nilai Kc, Ti, dan Td, digunakan grafik Ciancone *Set Point* dengan menggunakan Matlab sehingga diperoleh nilai Kc, Ti, dan Td masing-masing secara berturut-turut sebesar 2,76, 0,16, dan 0,012. Berikut grafik yang digunakan.



Gambar 9. Grafik Metode Ciancone

Dengan data tersebut, dapat dibuat blok diagram pengendali Ciancone sebagai berikut.

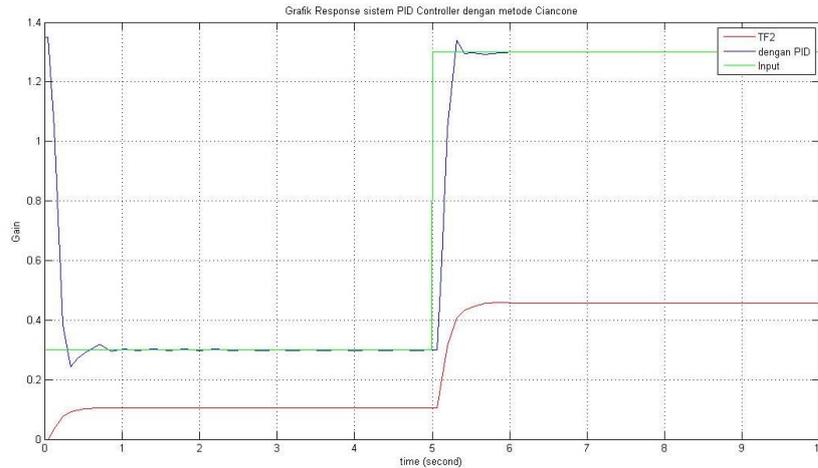


Gambar 10. Blok Diagram dengan PID controller Metode Ciancone

Pada blok PID Controller, digunakan Kp, Ki, dan Kd dengan menggunakan persamaan berikut.

$$K_p = K_c; K_i = \frac{K_c}{T_i}; K_d = K_c T_d \dots\dots\dots[11]$$

Oleh karena itu, diperoleh Kp, Ki, dan Kd masing-masing sebesar 2,76; 17,25; dan 0,033. Berikut adalah grafik hasil dari sistem blok Diagram PID controller dengan metode Ciancone.

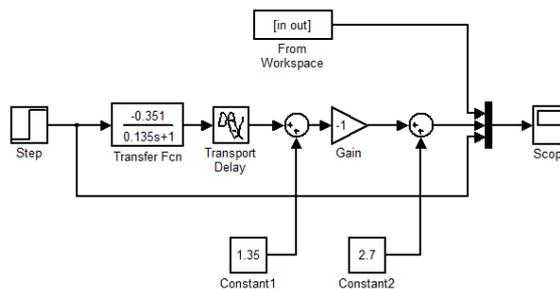


Gambar 11. Grafik hasil PID *Controller* metode Ciancone

### 3.3 Pemodelan Pengendali Sistem Metode Ciancone Modified

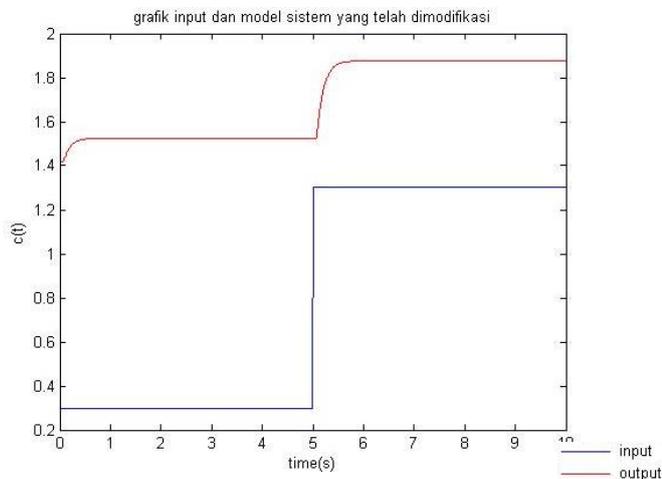
Pada Gambar 11 terlihat bahwa *initial state* grafik berlawanan dengan *input*. Hal ini berarti grafik pemodelan sistem tersebut masih bernilai negatif. Oleh karena itu, diperlukan modifikasi pada pemodelan sistem.

Untuk memodifikasi sistem, dilakukan pembalikan respon dengan mengalikan sistem dengan *gain* -1. Agar grafik dimulai dari nilai *initial value output* sistem asli, dilakukan penambahan sebesar dua kali besar respon. Berikut adalah blok diagram dengan metode *Ciancone modified*.



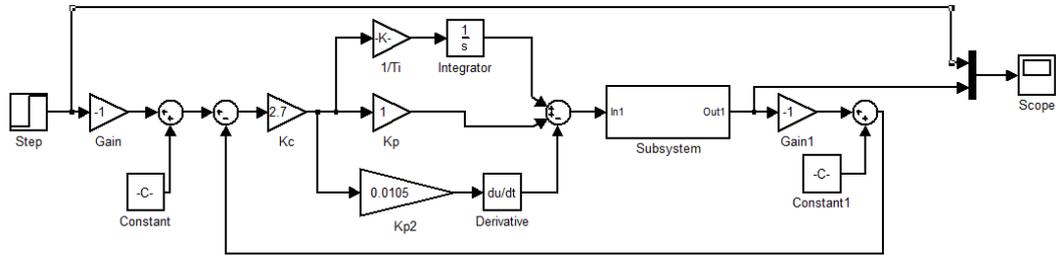
Gambar 12. Blok Diagram sistem yang telah dimodifikasi

Blok diagram pada Gambar 12 menghasilkan grafik sebagai berikut.



Gambar 13. Grafik *input* dan *output* blok diagram sistem yang telah dimodifikasi

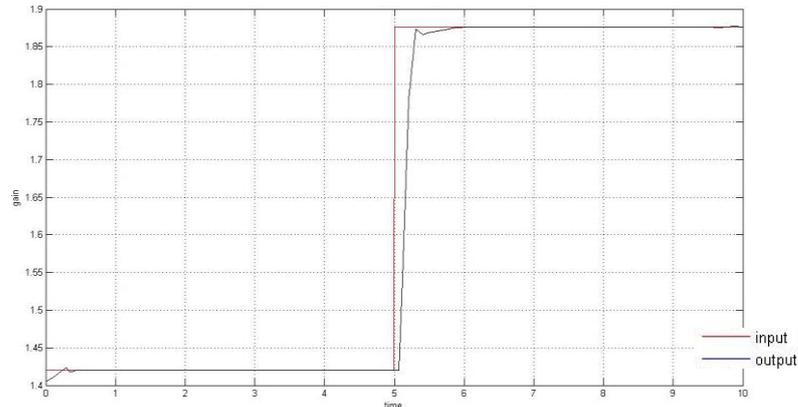
Pada Gambar 13, terlihat respon sistem masih belum dapat mengikuti *input* sehingga diperlukan desain pengendalian PID sistem.



Gambar 14. Blok diagram *simulink* lengkap dengan pengendaliannya

Pada Gambar 14, terlihat sistem dikali dengan *gain* -1 dan ditambah dengan konstanta. Hal ini bertujuan untuk memperoleh nilai *setpoint* yang sama dengan yang ada di sistem *Pressure Process Rig*.

Terlihat bahwa blok *controller* PID-nya sesuai dengan persamaan 10. Dengan menggunakan *mfile ciancone*, diperoleh konstanta  $K_c = 2,79$ ,  $T_i = 0,168$ , dan  $T_d = 0,0105$ . Berikut adalah grafik hasil dari respon pemodelan sistem dengan pengendali PID metode *Ciancone modified*:



Gambar 15. Grafik hasil

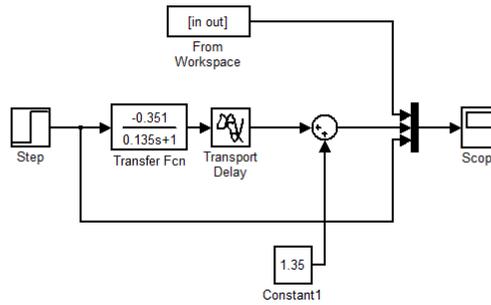
Tampak bahwa signal keluaran sudah mengikuti *setpoint* dengan baik. Pada Gambar 15, terlihat tidak terjadi *overshoot* untuk *transient response* dan *steady state error* bernilai mendekati nol. Hal ini menunjukkan bahwa pengendaliannya telah baik dan *valid*.

#### 4. HASIL PENELITIAN

Pemodelan Sistem *Pressure Process Rig* sebelumnya diperoleh fungsi alih sebagai berikut.

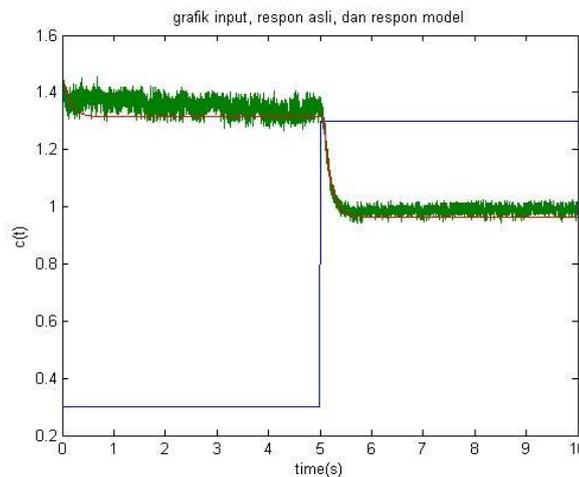
$$G(s) = \frac{Kp}{\tau s + 1} e^{-\tau s} = \frac{-0.351}{0.135s + 1} e^{-0.076s} \dots\dots\dots [12]$$

Selanjutnya, dilakukan simulasi dengan menggunakan *Simulink Matlab* untuk melihat hasil keluaran dari fungsi alih tersebut. Berikut ini adalah blok diagram yang digunakan.



Gambar 16. Blok Diagram awal

Pemodelan tersebut menggunakan *initial value* 0,3 dan *final value* 1,3. Berikut ini adalah grafik hasil simulasi.

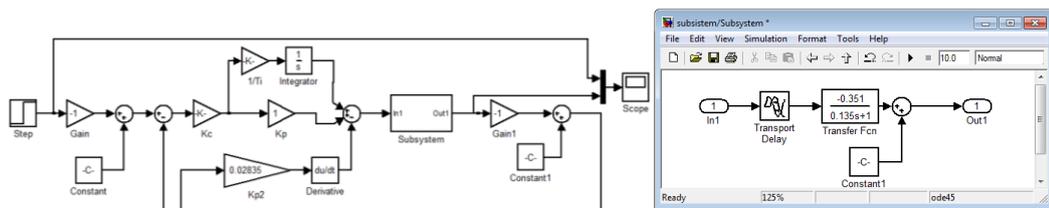


Gambar 17. Grafik hasil simulasi Blok Diagram awal

Dari grafik pada Gambar 17, terlihat bahwa grafik hasil pemodelan sistem telah mengikuti grafik respon asli sistem. Pada grafik hasil pemodelan sistem terdapat perbedaan nilai dengan grafik respon asli sistem pada keadaan *steady state*. Perbedaan ini terjadi akibat adanya *truncation error* pada saat pengambilan parameter pemodelan. Dengan grafik yang berhasil pada sistem asli, nilai yang diambil untuk dijadikan parameter pemodelan kurang akurat dan presisi sehingga menyebabkan perbedaan nilai pada saat kondisi *steady state*. Perbedaan ini cukup kecil sehingga dapat ditoleransi.

Desain Pengendali Sistem dengan metode Ciancone menggunakan grafik Ciancone untuk menentukan parameter  $K_c$ ,  $T_i$ , dan  $T_d$ . Parameter ini diperoleh dengan menggunakan nilai dari pemodelan sistem yang telah dilakukan. Dengan menggunakan mfile ciancone, diperoleh konstanta  $K_c = 2.79$ ,  $T_i = 0.168$ , dan  $T_d = 0.0105$ .

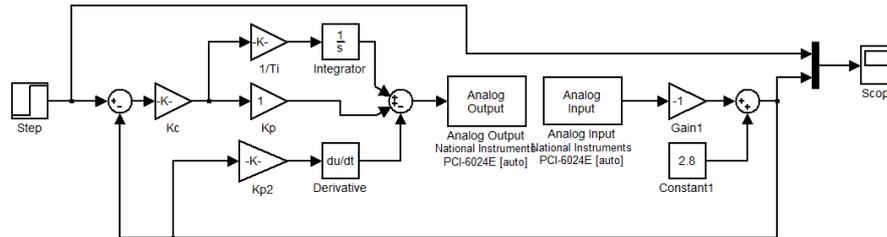
Berdasarkan persamaan tersebut, dilakukan pemodelan blok diagram pada Simulink Matlab. Berikut adalah blok diagram yang digunakan.



Gambar 18. Blok Diagram Sistem

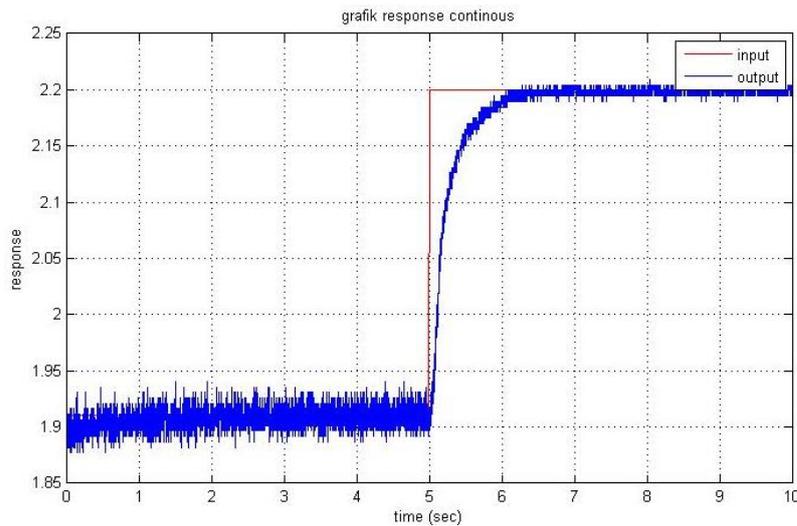
#### 4.1 Analisis Percobaan Pengendali PID Ciancone pada Sistem *Pressure Process Rig*

Pada percobaan Pengendali PID Ciancone pada *Real System Pressure Process Rig*, digunakan pengendali PID yang telah dimodelkan sebelumnya, yaitu dengan  $K_c = 2,79$ ,  $T_i = 0,168$ , dan  $T_d = 0,0105$ . Pengendali PID yang telah disimulasikan di Simulink Matlab dengan *plant* pemodelan sistem *Pressure Process Rig* di awal digunakan pada blok diagram yang langsung tersambung pada sistem asli *Pressure Process Rig*. Blok Diagram yang digunakan adalah sebagai berikut.



Gambar 19. Blok Diagram *Continuous* pada *Real Pressure Process Rig*

Dari percobaan tersebut, diperoleh hasil grafik sebagai berikut.



Gambar 20. Grafik percobaan *Continuous*

Dari grafik yang dihasilkan, terlihat bahwa response sistem mengikuti *input* yang diberikan. *Input* yang diberikan adalah *init step* dengan *initial condition* 1,91 dan *final value* sebesar 2,2 pada percobaan *continuous*. *Init step* memiliki *step time* 5 secon dan *sampling time* 0,001 untuk melihat *response* sistem yang lebih presisi.

Berdasarkan Gambar 22, terlihat bahwa *response* sistem telah berhasil mengikuti *input*. *Response* terlihat memiliki *rise time* yang cepat. *Response* juga terlihat tidak memiliki *overshoot* sehingga kerusakan sistem akibat *response* berlebih dapat dihindari. Pada grafik juga terlihat bahwa sistem langsung stabil tanpa membutuhkan *settling time* yang lama.

Sistem Pengendali PID Ciancone yang telah disimulasikan di Simulink Matlab dapat bekerja di sistem *Pressure Process Rig*.

## 5. KESIMPULAN

Dari percobaan yang telah dilakukan, diperoleh kesimpulan sebagai berikut.

- a. *Pressure Process Rig* dapat dimodelkan dengan persamaan orde 1 menggunakan *Process Reaction Curve*.
- b. Pada orde 1, sistem dapat dikendalikan menggunakan PID dengan parameter  $K_c = 2,79$ ,  $T_i = 0,168$ , dan  $T_d = 0,0105$  dengan menggunakan metode ciancone.
- c. Sistem dapat mengikuti *set point* yang diberikan.
- d. Desain pengendali PID dapat dimodelkan dengan metode ciancone.
- e. Untuk mendapatkan pengendali PID yang sesuai dengan keluaran, *set point* yang masuk ke sistem perlu dimodifikasi menjadi terbalik.
- f. Berdasarkan grafik, *set point* hasil modifikasi berhasil menggantikan *set point* sebelumnya.
- g. Pengendali PID *Continous* yang telah dimodelkan pada saat simulasi dapat digunakan pada *plant* asli, yaitu *Pressure Process Rig*. Keluaran sistem *Pressure Process Rig* mengikuti *setpoint* yang diberikan. Pengendali PID Ciancone telah berhasil dilakukan.

## REFERENSI

- [1]. Ashok Kumar, Rajbir Morya, Munish Vashishath. "Performance Comparison Between Various Tuning Strategies: Ciancone, Cohen Coon & Ziegler- Nicholas Tuning Methods." *International Journal of Computers & Technology*, 2013: 60.
- [2]. D.E. Seborg, T.F. Edgar, D.A. Mellichamp. 1989. *Process Dynamics and Control*. New York: Wiley.
- [3]. Marlin. 2000. *Process Control: Designing Process and Control Systems for Dynamic Performance*. 2nd Ed. New York: McGraw-Hill.
- [4]. Nise, Norman S. 2004. *Control Systems Engineering*. 4th Ed. New York: John Wiley.
- [5]. Ogata, Katsuhiko. 1991. *Modern Control Engineering*. 2nd Ed. Jakarta: Erlangga.
- [6]. Pratomo, Vector Anggit. "Perancangan Pengendali PID Pada *Pressure Process Rig* (38-714) Berbasis Mikrokotroller AVR Atmega8535." *Jurnal Teknik FTUP*, 2012: 106-113.
- [7]. R. Ciancone, T. Marlin. "Tune Controllers to meet Plant Objectives." *Control*, 1992: 50-57.
- [8]. Shinskey, F.G. 1996. *Process-Control Systems, Application, Design, and Tuning*. 4<sup>th</sup> Edition.. New York: McGraw-Hill.