

# Kontrol PID Untuk Pengaturan Kecepatan Motor DC Dengan Metode *Tuning Direct Synthesis*

Handy Wicaksono<sup>1</sup>, Josaphat Pramudijanto<sup>2</sup>

<sup>1</sup>Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Kristen Petra  
e-mail : [handy@peter.petra.ac.id](mailto:handy@peter.petra.ac.id)

<sup>2</sup>Fakultas Teknologi Industri, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya

## Abstrak

Kontroler PID adalah kontroler yang sampai sekarang masih banyak digunakan di dunia industri. Hal yang krusial pada desain kontroler PID ini ialah menentukan parameter kontroler atau tuning. Dari banyak metode tuning yang telah dikembangkan saat ini, akan dibahas metode tuning Direct Synthesis.

PLC yang umumnya digunakan sebagai alat pengatur urutan bisa dimanfaatkan sebagai kontroler PID digital dengan memanfaatkan modul ASCII (Omron) yang bisa mengadaptasi pemrograman dalam bahasa BASIC. Dengan mengintegrasikan PLC sebagai kontroler PID dan motor DC sebagai plant, metode tuning Direct Synthesis dapat diimplementasikan.

Dari hasil eksperimen, terbukti bahwa penggunaan kontroler PID dengan metode tuning Direct Synthesis untuk pengaturan kecepatan motor DC memberikan perbaikan kriteria performansi pada plant yang signifikan jika dibandingkan dengan plant tanpa kontroler.

**Kata kunci** : kontroler PID, tuning, PLC.

## Abstract

*PID controller is a well known controller which has been used widely in the industry. The significant thing in the design of PID controller is tuning of its parameters. From many tuning methods which has been developed today, Direct Synthesis tuning method is chosen for this experiment.*

*Generally, PLC is used to control the sequence. But we can use it as digital PID controller too by using ASCII module (Omron) which can adapt with BASIC. By integrating PLC as PID controller and DC motor as the plant, Direct Synthesis tuning method can be implemented in that plant.*

*From the experiment's result, it has been proved that by using PID controller with Direct Synthesis tuning method, the plant's performance criteria has been improved significantly.*

**Keywords** : PID controller, tuning, PLC BD12.

## 1. Pendahuluan

Kontroler PID adalah kontroler berumpanbalik yang paling populer di dunia industri. Selama lebih dari 50 tahun, kontroler PID terbukti dapat memberikan performa kontrol yang baik meski mempunyai algoritma sederhana yang mudah dipahami [1]. Hal krusial dalam desain kontroler PID ialah *tuning* atau pemberian parameter P, I, dan D agar didapatkan respon sistem yang diinginkan.

Salah satu metode yang muncul ialah *tuning* berdasar model *plant*, karena identifikasi plant bukan lagi hal yang sulit untuk dilakukan. Salah satu jenisnya ialah *Direct Synthesis* yang memerlukan model *plant* sebenarnya dan model *plant* yang diinginkan untuk mendapatkan parameter P, I, D dari kontroler. [2].

Sementara itu, di dunia industri juga dikenal adanya *Programmable Logic Controller* (PLC) sebagai alat pengatur urutan proses secara digital. Namun sekarang ini PLC telah dapat juga menangani proses analog. PLC C200H OMRON mengadaptasi hal itu dengan munculnya *special unit* seperti *Analog Input Unit*, *Analog Output Unit*, *PID Controller*, *ASCII Unit*, dan lain – lain [3].

---

**Catatan**: Diskusi untuk makalah ini diterima sebelum tanggal 1 Juni 2004. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 4, nomor 2, September 2004.

Karena itu, penulis akan mengimplementasikan kontroler PID pada modul ASCII untuk mengatur kecepatan motor DC. Selain itu akan dilakukan penerapan metode *tuning Direct Synthesis* pada kontroler PID. Sebagai catatan, tidak semua metode *tuning* cocok digunakan untuk jenis-jenis *plant* tertentu. Misalnya: penggunaan metode *tuning* Ziegler-Nichols di Laboratorium Sistem Pengaturan Unibraw untuk pengaturan posisi motor DC justru memberikan hasil yang mengecewakan saat kontroler PID diterapkan [4].

## 2. Dasar Teori

### 2.1 Kontroler PID [4]

Kontroler adalah komponen yang berfungsi meminimasi sinyal kesalahan. Tipe kontroler yang paling populer ialah kontroler PID. Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar.

Persamaan kontroler PID dalam bentuk Laplace:

$$M(s) = K_c \left( E(s) + \frac{1}{T_i s} E(s) + T_d s E(s) \right) \quad (1)$$

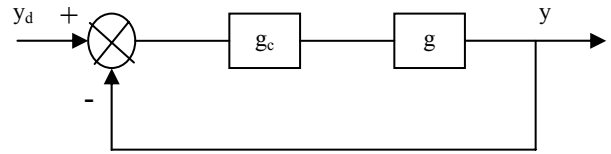
### 2.2 Tuning Kontroler dengan Model *Plant* Nyata [2]

Aspek yang sangat penting dalam desain kontroler PID ialah penentuan parameter kontroler PID supaya sistem *close loop* memenuhi kriteria performansi yang diinginkan. Hal ini disebut juga dengan *tuning* kontroler.

Seiring dengan berkembangnya penelitian tentang identifikasi suatu sistem “*black box*”, maka memperoleh *transfer function* atau karakteristik dari sistem tersebut bukanlah hal yang teramat sulit. Hal ini menyebabkan metode *tuning* kontroler yang membutuhkan model *plant* sebenarnya juga dapat dilakukan dengan relatif mudah, misalnya dengan metode *Direct Synthesis*.

Metode ini terlebih dulu menentukan perilaku output yang diinginkan (*reference*) dengan membuat bentuk trayektorinya, dan model prosesnya (*plant*) digunakan untuk secara langsung mendapatkan persamaan kontroler yang sesuai. Berikut ini penurunan rumusnya.

Jika diketahui diagram blok dari suatu sistem ialah sebagai berikut.



Gambar 1. Blok diagram dari sistem kontrol berumpan balik

Maka *closed-loop transfer function* ialah sebagai berikut :

$$y(s) = \frac{gg_c}{1 + gg_c h} y_d(s) \quad (2)$$

Dan pendekatan yang diinginkan untuk mendapatkan *setpoint* yang baru dimodelkan dengan trayektori yang diinginkan berikut :

$$\frac{y(s)}{y_d(s)} = q(s) = \frac{gg_c}{1 + gg_c h} \quad (3)$$

sehingga persamaan kontrolernya :

$$g_c = \frac{1}{g} \left( \frac{q}{1-q} \right) \quad (4)$$

Sesuai dengan *transfer function plant* motor DC yang telah didapatkan dan berbentuk *First Order Plus Dead Time (FOPDT)*, yaitu:

$$g(s) = \frac{K.e^{-\alpha s}}{\tau s + 1} \quad (5)$$

Dan dipilih *reference trajectory*:

$$q(s) = \frac{e^{-\alpha_r s}}{\tau_r s + 1} \quad (6)$$

Dengan memasukkan hasil Persamaan 5 dan Persamaan 6 pada Persamaan 4 maka didapat persamaan kontroler :

$$g_c = \frac{(\tau s + 1)}{K} \left( \frac{1}{\tau_r s + 1 - e^{-\alpha s}} \right) \quad (7)$$

Akhirnya didapatkan kontroler dalam bentuk persamaan, namun untuk merealisasikannya sangat sulit karena besaran tidak bisa diimplementasikan dalam komponen analog. Namun dengan adanya implementasi kontroler PID pada mikroprosesor dan komputer digital membuat besaran tersebut bisa diimplementasikan. Melalui model dasar kontroler ini didapatkan beberapa macam nilai *tuning* PID yang berbeda-beda.

Dengan menggunakan pendekatan Pade orde 1 :

$$e^{-\alpha s} \approx \frac{1 - \frac{\alpha}{2}s}{1 + \frac{\alpha}{2}s} \quad (8)$$

Pada Persamaan 7, kontroler yang didapatkan menjadi

$$g_c = \frac{\tau}{K(\tau_r + \alpha)} \left( 1 + \frac{1}{\tau s} \right) \left( \frac{1 + \frac{\alpha}{2}s}{1 + \tau^* s} \right) \quad (9)$$

dengan  $\tau^*$  adalah *filter* yang mempunyai persamaan sebagai berikut:

$$\tau^* = \frac{\alpha \tau_r}{2(\alpha + \tau_r)} \quad (10)$$

Persamaan 9 mempunyai struktur sesuai dengan struktur kontroler PID komersial. Maka parameter kontroler PID komersial dapat dicari sebagai berikut :

$$K_c = \frac{\tau}{K(\tau_r + \alpha)} ; \quad \tau_I = \tau ; \quad \tau_D = \frac{\alpha}{2} ;$$

$$\tau^* = \frac{\alpha}{2} \left( \frac{\tau_r}{\alpha + \tau_r} \right) \quad (11)$$

Dengan kontroler yang sama, persamaan di atas dapat disusun kembali menjadi :

$$g_c = \frac{\tau + \frac{\alpha}{2}}{K(\tau_r + \alpha)} \left[ 1 + \frac{1}{(\tau + \frac{\alpha}{2})s} + \left( \frac{\alpha \tau}{\tau + \frac{\alpha}{2}} \right) \left( \frac{1}{\tau^* s + 1} \right) \right] \quad (12)$$

Maka parameter *tuning* dari kontroler PID ideal ialah :

$$K_c = \frac{\tau + \frac{\alpha}{2}}{K(\tau_r + \alpha)} \quad \tau_I = \tau + \frac{\alpha}{2} \quad \tau_d = \frac{\frac{\alpha}{2} \tau}{\tau + \frac{\alpha}{2}}$$

$$\tau^* = \frac{\alpha}{2} \left( \frac{\tau_r}{\alpha + \tau_r} \right) \quad (13)$$

Parameter inilah yang digunakan dalam eksperimen.

### 3. Perancangan Model

#### 3.1 Perencanaan Hardware [3]

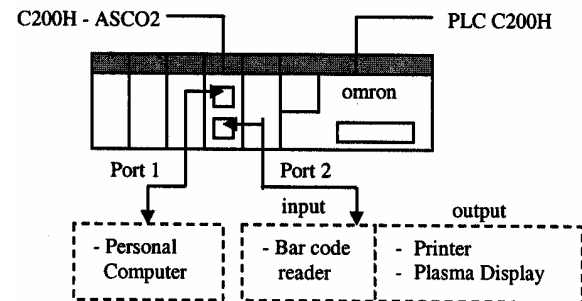
Keperluan *hardware* meliputi : modul ASCII pada PLC sebagai alat kontrol utama, modul Analog Input pada PLC dan *hardware* pendukung (Digital to Analog Converter, Amplifier) untuk mengkondisikan sinyal antara PLC dan *plant*. Hanya unit ASCII yang akan dijelaskan dengan detail pada makalah ini.

#### 3.1.1 Unit ASCII [5]

Unit ASCII adalah unit pelengkap cerdas dari PLC C200H OMRON yang membuat sistem kontrol berbasis PLC lebih fleksibel dan berkemampuan tinggi. Unit ASCII ini dapat digunakan untuk memonitor sistem, memproses data, membuat laporan dan mengerjakan tugas – tugas lainnya. Pemrograman pada ASCII Unit dikerjakan dengan BASIC, sebagai pengganti ladder diagram, sehingga lebih cocok untuk memproses data analog.

#### Konfigurasi

Berikut ini konfigurasi sistem dengan modul ASCII.



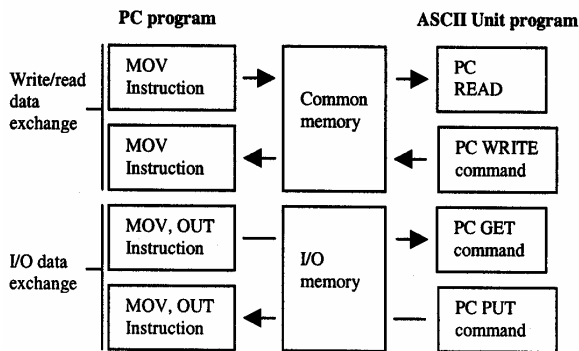
Gambar 2. Konfigurasi Sistem dengan Modul ASCII

#### Komunikasi

Untuk menggunakan Unit ASCII yang berhubungan dengan PLC, diperlukan program untuk Unit ASCII yang ditulis dalam BASIC. Perintah pertukaran data harus disertakan ke dalam program PLC kecuali jika pernyataan perintah yang digunakan telah menggunakan petunjuk daerah memori yang spesifik (misal : PC READ "@...", PC WRITE "@..."). Perintah tersebut harus menentukan jumlah *word* yang akan ditransfer, *base address*, dan daerah memori yang spesifik. Hal ini bisa dilakukan dengan menggunakan instruksi PC MOV.

Ada 2 cara Unit ASCII dapat berkomunikasi dengan PLC. Pada metode pertama, PLC mengontrol *timing* transfer data antara 2 alat ini. ASCII Unit "meminta" akses ke daerah memori data PLC dengan menggunakan *statement* PC READ, PC WRITE, PC GET, atau PC PUT, dan kemudian menunggu PLC merespon dengan menyalakan *read* atau *write flag*.

Pada metode yang ke dua, tidak ada kode pertukaran data khusus dari PLC yang diperlukan untuk mengkomunikasikan 2 alat ini. Jika parameter penunjuk daerah memori telah ditentukan dengan *statement* PC READ atau PC WRITE, Unit ASCII dapat langsung mengakses daerah memori PLC yang telah ditentukan. Gambar – gambar berikut ini mengilustrasikan hubungan antara program di PLC dan program di ASCII Unit.

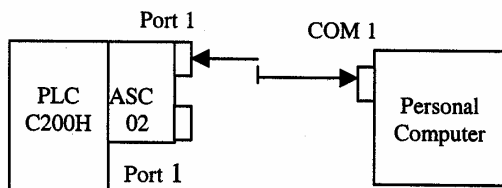


Gambar 3. Hubungan antara program di PLC dan program di unit ASCII

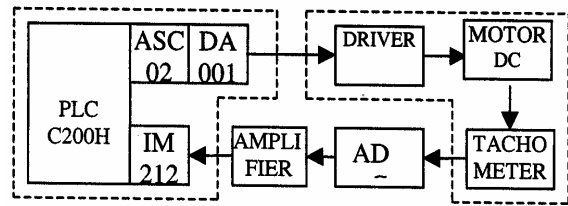
Program BASIC untuk ASCII Unit harus ditulis pada PC yang dihubungkan dengan *port 1* ASCII Unit melalui RS 232-C. Sebuah program dapat ditransfer ke ASCII Unit dari PC atau alat penyimpan lain dengan perintah LOAD. LOAD juga digunakan untuk mentransfer program dari EEPROM ke RAM dalam Unit ASCII.

Sebaliknya, program dapat ditransfer dari RAM ke EEPROM dari ASCII Unit atau ke PC yang terhubung dengan perintah SAVE. Selain itu, program juga dapat ditransfer dengan mudah dengan *software* bawaan dari OMRON yaitu SYSMATE ASCII. ASCII Unit dihubungkan ke alat *peripheral* melalui dua RS-232C *interface*. Konektor dB 9 digunakan untuk kedua *port*.

Berikut ini susunan rangkaian untuk *upload* program.



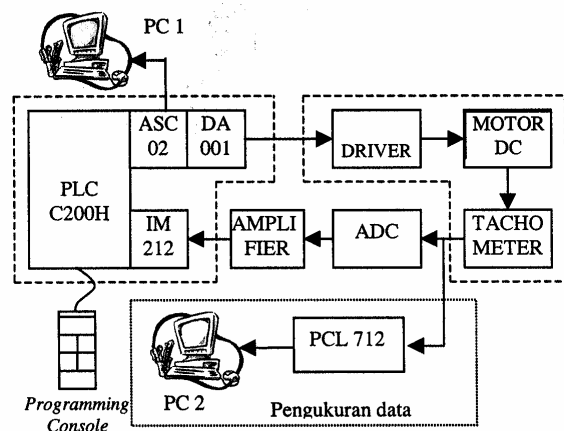
Gambar 4. Rangkaian *hardware* untuk *upload* output program



Gambar 5. Rangkaian *hardware* untuk *running* program

### 3.1.2 Rangkaian Lengkap *Hardware*

Gambar 6 menunjukkan rangkaian lengkap PLC, *plant*, dan *hardware* pendukung lainnya yang dipergunakan dalam percobaan



Gambar 6. Rangkaian *hardware*

### 3.2 Perencanaan *Software*

Selain masalah *hardware*, *software* yang bekerja sebagai “jantung” dari sistem sangat penting untuk direncanakan dengan tepat. Di antaranya ialah algoritma untuk transfer data dan implementasi kontroler digital.

#### 3.2.1 Algoritma Transfer Data [5]

Data dari *plant* (motor DC) berupa tegangan yang dihasilkan oleh *tachometer*. Tegangan analog antara 0 – 5 V tersebut dimasukkan ke dalam ADC 8 bit dan akan diubah menjadi 8 digit bilangan biner yang merepresentasikan nilai tegangan analog tersebut berdasarkan nilai – nilai biner dari MSB (*Most Significant Bit*) sampai LSB (*Least Significant Bit*). Nilai keluaran ADC sebesar 0 – 5 V tersebut akan dikuatkan sebesar 4 kali karena *level logic* pada *Input Module* PLC adalah 0 - 24 V, yang akan dianggap sebagai data input PLC.

Data input PLC ini dengan *ladder* tertentu akan dikirimkan kepada modul ASCII, dimana modul ini akan menerima data dengan program BASIC tertentu. Di dalam modul ASCII, data akan diolah sesuai keinginan *programmer* (dalam hal ini dimasukkan dalam program kontroler PID), kemudian hasil akhirnya akan dikirimkan lagi ke PLC dengan program BASIC tertentu. PLC akan menerima data dengan *ladder* tertentu juga.

Sedangkan data dari PLC berupa data digital 8 bit yang akan diubah ke dalam bentuk tegangan analog melalui modul analog output (DA 001).

**3.2.2 Implementasi Algoritma Kontroler Digital[6]**

Selain itu implementasi kontroler PID secara digital juga harus diterapkan dengan benar dan teliti. Modul ASCII pada PLC akan digunakan sebagai kontroler PID. Aspek-aspek implementasi kontroler dalam program (digital) ialah sebagai berikut:

- Persamaan kontroler PID digital :

$$m_n = K_c \left[ T_d \frac{(e_n - e_{n-1})}{\Delta t} + e_n + \frac{1}{T_i} \sum_{k=0}^n e_k \cdot \Delta t \right] \quad (14)$$

- Penggunaan *Sampling time*: 0,2 detik.
- Perlu adanya penambahan konstanta MV yang mewakili nilai dari *manipulated variable* saat *steady state* supaya tidak terjadi keadaan sinyal kontrol = 0 saat *error* = 0. Besar MV = 3,5 V.
- Mengatasi *integral windup* atau *integral saturation* dengan membatasi besarnya komponen integral atau sinyal kontrol yaitu antara 2,5 – 8 V.

Dengan memasukkan nilai – nilai parameter kontroler PID berdasar metode *tuning* : *Direct Synthesis* yang telah dijelaskan, berikut hasil yang didapatkan.

$$K_p = 1,826 \quad T_i = 1,632 \quad T_d = 0,028$$

**4. Hasil Pengukuran dan Analisa**

**4.1 Identifikasi Sistem [7]**

Identifikasi proses atau sistem dilakukan berdasarkan data percobaan/eksperimen dengan mengukur sinyal masukan dan keluaran. Identifikasi yang dilakukan berikut adalah metode identifikasi Strejc yang merupakan salah satu contoh dari metode eksperimental. Dari perhitungan yang dilakukan berikut model matematika yang didapatkan:

$$G(s) = \frac{0,847}{(1,604s + 1)(0,056s + 1)} \quad (15)$$

Dari perhitungan dengan Matlab untuk membandingkan hasil simulasi dan respon sebenarnya, didapatkan *norm error* yang cukup kecil antara keduanya yaitu sebesar : 4,8608.

**4.2 Kriteria Performansi [8]**

Dengan mengacu pada pengertian yang diberikan Ogata [8], berikut ini kriteria performansi dari *plant* yang digunakan:

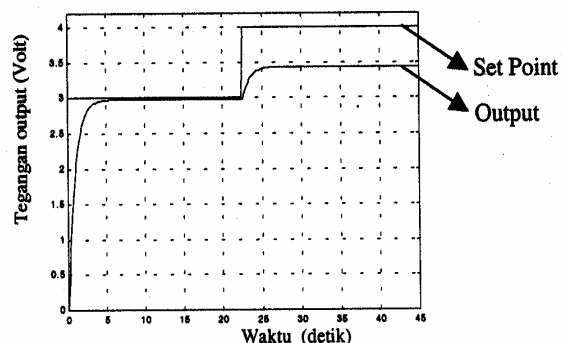
- Berada dalam pita akurasi 5% (lebih atau kurang dari 5% *set point*). Dalam besaran tegangan : 3,8 V – 4,2 V.
- *Error steady state* kurang dari 5% dari *set point* (agar berkorelasi dengan syarat akurasi). Dalam besaran tegangan kurang dari 4,2 V.
- *Rise time* maksimal 1 s.
- *Setting time* maksimal 2 s
- *Maximum overshoot* sebesar 5% dari *set point*. Dalam besaran tegangan : maksimal 4,2 V.

**4.3 Hasil dan Analisa Pengujian Sistem**

Pengukuran data dilakukan dengan menyusun rangkaian seperti pada gambar 6, dan merekam hasilnya.

**4.3.1 Simulasi dan Percobaan Close Loop Test I-Perubahan Set Point**

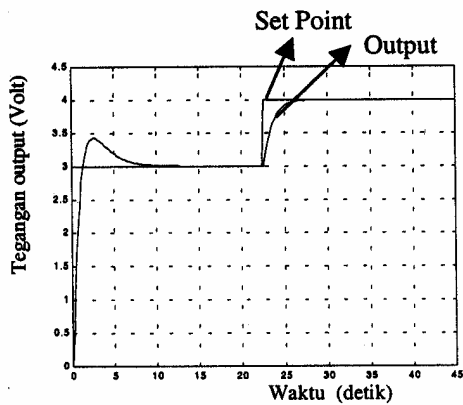
Eksperimen pertama yang dilakukan ialah dengan memberikan perubahan *set point* pada *plant* yang telah *running* dan *steady*. Berikut ini gambar hasil simulasi dengan MATLAB dari eksperimen tersebut untuk *plant* tanpa kontroler.



Gambar 7. Simulasi *close loop tes I* untuk *plant* tanpa kontroler

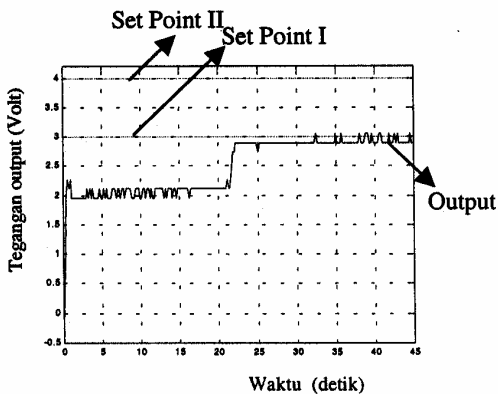
Sedang gambar hasil simulasi dengan MATLAB dari eksperimen *plant* dengan kontroler PID

menggunakan metode *tuning Direct Synthesis* ialah :



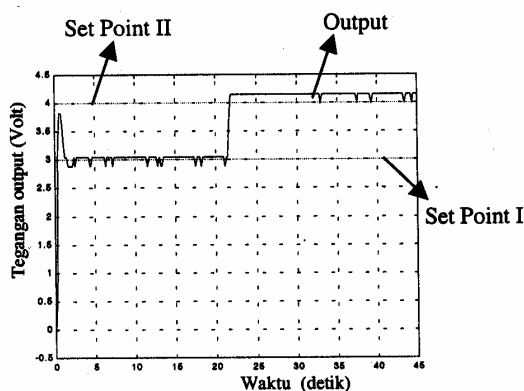
Gambar 8. Simulasi *close loop test I* untuk *plant* dengan kontroler PID metode *tuning Direct Synthesis*

Sedang hasil eksperimen sebenarnya dari *plant* tanpa kontroler ialah sebagai berikut :



Gambar 9. *Close loop test I* untuk *plant* tanpa kontroler

Untuk *plant* dengan kontroler PID-*Direct Dynthesis*:



Gambar 10. *Close loop test I* untuk *plant* dengan kontroler PID metode *tuning Direct Synthesis*

Berikut ini hasil pencatatan kriteria performansi masing-masing sistem :

Tabel 1. Hasil pengamatan kriteria performansi pada *close loop test I*

No	Metode	Akurat*	Error SS. (V)	Max. Over shoot (%)	Rise Time (det.)	Settling Time (det.)
1	Tanpa kontroler (simulasi)	Tidak	0,56	-	1,26	-
2	Direct Synthesis (simulasi)	Ya	0	-	2,52	4,32
3	Tanpa kontroler (percobaan)	Tidak	1,106	-	0,54	-
5	Ideal Controller (percobaan)	Ya	0,144	-	0,36	0,54

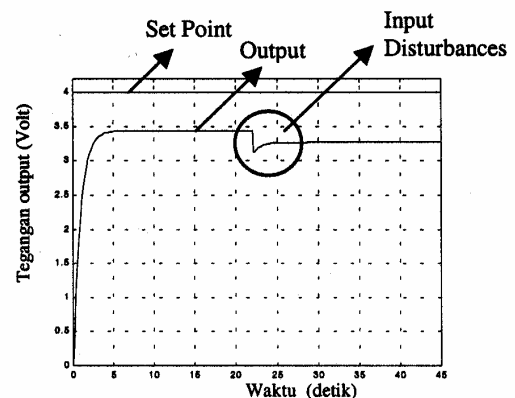
\*Akurat: berada dalam pita akurasi 5%

Dari tabel di atas nampak bahwa penambahan kontroler PID metode *tuning Direct Synthesis* memberikan perbaikan pada komponen-komponen kriteria performansi seperti tersebut di atas, dan hasilnya sistem tersebut memenuhi kriteria performansi.

#### 4.3.2 Simulasi dan Percobaan *Close Loop Test II-Input Disturbances*

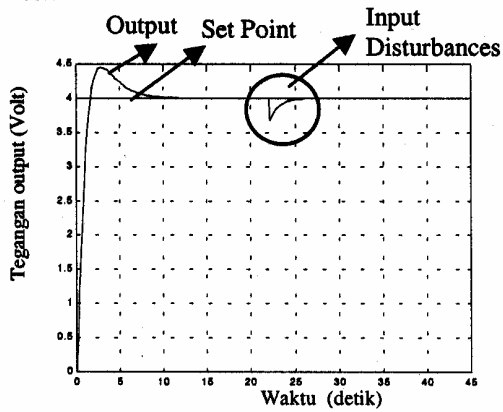
*Close loop test* ini dilakukan dengan memberikan tegangan input sebesar 4 V sehingga motor akan bergerak dan mencapai keadaan *steady*, kemudian akan diberikan gangguan (*disturbances*) berupa rem magnetik. Dengan percobaan ini akan dilihat bagaimana aksi kontroler dalam menangani adanya gangguan ini.

Hasil simulasi *plant* tanpa kontroler:



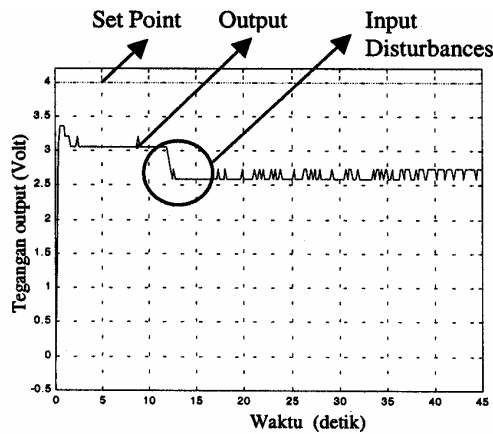
Gambar 11. Simulasi *close loop test II* untuk *plant* tanpa kontroler

Sedang hasil simulasi *plant* dengan kontroler PID-*Direct Synthesis*:



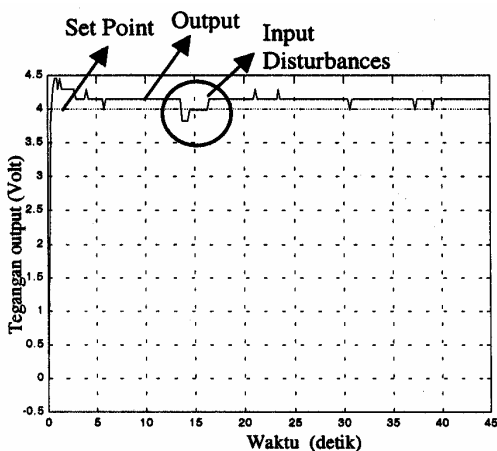
Gambar 12. Simulasi *close loop test II* untuk *plant* dengan kontroler PID metode *tuning Direct Synthesis*

Hasil percobaan sebenarnya dari *plant* tanpa kontroler:



Gambar 13. *Close loop testII* untuk *plant* tanpa kontroler

Sedang hasil percobaan sebenarnya dari *plant* dengan kontroler PID-*Direct Synthesis* :



Gambar 14. *Close Loop Test II* untuk *plant* dengan kontroler PID metode *tuning Direct Synthesis*

Berikut ini hasil pencatatan kriteria performansi masing-masing sistem :

Tabel 2. Hasil pengamatan waktu kembali pada *close loop test II*

No.	Metode	Waktu Kembali (detik)
1	Tanpa kontroler (simulasi)	-
2	Dengan kontroler PID (simulasi)	2,88
3	Tanpa kontroler (percobaan)	-
4	Dengan kontroler PID (percobaan)	2,88

Dari tabel di atas nampak bahwa penambahan kontroler PID metode *tuning Direct Synthesis* memberikan perbaikan waktu kembali pada *plant*.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari pembahasan dalam penelitian ini dapat disimpulkan bahwa :

1. Implementasi kontroler PID pada modul ASCII-PLC C200H untuk mengatur kecepatan motor DC dapat dilakukan.
2. Implementasi metode *tuning Direct Synthesis* pada kontroler PID memberikan kriteria performansi *plant* yang cukup baik (memberi akurasi, *error steady state* : 0,144 detik, *maximum overshoot* : 0, *rise time* : 0,36 detik, dan *settling time* : 0,54 detik) dan kemampuan untuk kembali mencapai *set point* saat diberikan *disturbance*.

### 5.2 Saran

Saran – saran untuk mengembangkan penelitian ini ialah:

1. Implementasi *self-tuning* PID dengan *fuzzy logic* untuk peningkatan performa kontrol dari kontroler PID.
2. Pemilihan jenis dan karakteristik *plant* yang tepat agar berbagai pengembangan dalam eksperimen dapat dilakukan.

## Daftar Pustaka

- [1] Willis, M. J., 1999. Proportional-Integral-Derivative Control, <URL:<http://lorien.ncl.ac.uk/ming/pid/pid.pdf>>.
- [2] Ogunnaike, B.A., dan Ray, W.H.1994. Process Dynamics, Modelling and Control. Oxford University Press . New York, USA .
- [3] ....., 1988. Sysmac Programmable Controller C200H Operation Manual, OMRON.
- [4] Chairuzzaini dkk., 1998. Metode Ziegler-Nichols pada Sistem Kontrol Nichols pada Perancangan Kontroler PID, <URL:<http://www.elektroindonesia.com/elektro/tutor12.html>>.
- [5] ....., 1992. C200H-ASC02 ASCII Unit, OMRON.
- [6] Bennet S. 1988. Real Time Computer Control: An Introduction. Prentice Hall Inc.
- [7] Annexe, 1998. Methodes d'identification De Strejc, <URL:<http://www.univlille1.fr/eudil/belk/pral.htm>>
- [8] Ogata, Katsuhiko. 1994. Teknik Kontrol Automatik I. Penerbit Erlangga.
- [9] Hutomo, Hadi. 1995. Perencanaan dan Pembuatan Fuzzy Logic Unit dengan Memakai ASCII Unit pada PLC C200H dan Salah Satu Aplikasinya. STTS
- [10] Smith, C.A., dan Corripio, A.B. 1997. Principles And Practice of Automatic Process Control . John Wiley and Sons Inc . USA
- [11] ....., 1988. Sysmac C200H Analog I/O Units Operation Guide, OMRON.