

# PERANCANGAN SISTEM KONTROL KECEPATAN MOTOR DC DENGAN PID LABVIEW 2010

Fauzy<sup>1)</sup>, Sugiatmo Kasmungin<sup>2)</sup>

<sup>1, 2)</sup>Jurusan Teknik Perminyakan, Universitas Trisakti, Jakarta 11440,  
email: [fauzy.said1107@gmail.com](mailto:fauzy.said1107@gmail.com)

## Abstract

*The stability of the system is the main objective for Control System Design criteria. PID control with closed loop system is the one of control which can improve the stability of the system from disturbance effect. In this study, DC Motor Speed Control is designed with PID LabVIEW 2010 Software and Proximity Sensor as a closed loop feedback. The system allows the operator to control speed of DC Motor from LabVIEW front panel which automatically stabilized when load –which usually slow down the speed of DC Motor, applied. PID LabVIEW 2010 generates voltage output 0-5Vdc and drives DC Motor through analog output channel from NIDAQ 600, which received by Non-Inverting Amplifier and amplified to 0-12Vdc output. PID Control with Closed Loop system may dampen the speed response second order of DC Motor to have value of overshoot number  $\approx 0$  with steady state error of  $\pm 2\%$ . The system feedback can stabilize this condition from outside interference. The result of this speed control system has success criteria of overshoot ( $M_p$ ) = 0 and Error Steady State ( $E_{ss}$ ) = 0.82%.*

**Keywords:** PID Control, PID LabVIEW 2010, NIDAQ 6008, DC Motor

## Abstrak

*Dalam mendesain sistem kontrol, stabilitas sistem merupakan hal yang sangat penting. Keberadaan sistem kontrol umpan balik PID memungkinkan kestabilan sistem dapat diraih meskipun terdapat gangguan pada sistem. Dalam penelitian ini, penulis merancang sebuah sistem kontrol kecepatan motor DC. Pada sistem ini digunakan PID LabVIEW 2010 sebagai sistem kontrol dan Sensor Proximity sebagai input dari proses variabel PID LabVIEW 2010. Sistem ini dapat dioperasikan oleh operator melalui front panel LabVIEW. Operator dapat mengatur kecepatan Motor DC yang kemudian kecepatan ini akan dijaga tetap sama oleh sistem kontrol ini meskipun ada pembebanan/gangguan dari luar sistem yang mengakibatkan penurunan kecepatan. PID LabVIEW 2010 ini memberikan keluaran output tegangan 0-5Vdc melalui kanal analog output dari NIDAQ 6008. Tegangan ini akan diterima oleh Penguat Non-Inverting untuk dikuatkan menjadi 0-12Vdc. Tegangan ini kemudian dijadikan sebagai tegangan input dari Motor DC. Sistem kontrol PID dengan umpan balik ini dapat membuat respon motor DC (respon sistem orde 2) dapat diredam sehingga overshoot  $\approx 0$  dan error steady state  $\pm 2\%$ . Sedangkan umpan balik dapat membuat kondisi sistem ini stabil meskipun adanya gangguan dari luar. Hasil rancangan sistem kontrol kecepatan ini memenuhi kriteria yang diharapkan dengan overshoot,  $M_p = 0$  dan Error Steady State,  $E_{ss} = 0,82\%$ .*

**Kata Kunci:** Kontrol PID, PID LabVIEW 2010, NIDAQ 6008, Motor DC

## 1. PENDAHULUAN

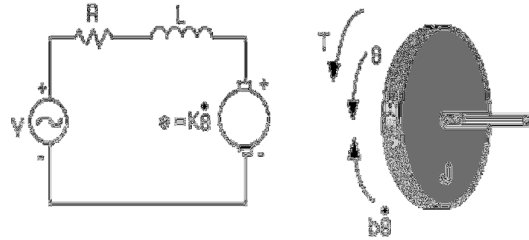
Berkembangnya teknologi tidak terlepas dari kemajuan bidang elektronika khususnya dalam Sistem Kontrol Analog. Salah satu aktuator yang sering dipergunakan dalam proses automasi adalah Motor DC. Dalam pemanfaatannya, Motor DC dalam berbagai hal, seperti: robot pemindah barang, konveyor pada perusahaan manufaktur dan lain-lain. Setiap motor DC memiliki respon sistem yang berbeda-beda ketika diberikan sebuah inputan sehingga diperlukan sebuah sistem kontrol dapat membuat respon dari motor DC. ini khususnya kecepatannya dapat dikontrol secara stabil walaupun terdapat gangguan dari luar sistem itu sendiri. Pada Tugas Akhir ini, PID LabVIEW digunakan sebagai sistem kontrol otomatis untuk mengontrol kecepatan dari motor agar tidak terjadi kondisi overshoot dan error steady state yang kecil dengan motor DC sebagai objeknya.

Sensor yang digunakan adalah sensor *proximity* untuk menghitung banyaknya putaran dari motor DC yang akan ditampilkan pada *Man Machine Interface* (MMI) sebagai representasi dari frekuensi motor DC tersebut. Sistem kontrol yang digunakan dalam sistem ini adalah PID *LabVIEW* dengan *software Lab VIEW 2010*.

## 2. LANDASAN TEORI

### 2.1 Motor DC dan Generator DC

Aktuator yang biasa digunakan dalam sistem kontrol adalah motor DC. Berikut ini adalah rangkaian listrik dari armatur dan mekanisme dari rotor seperti diperlihatkan pada gambar 2.1.



Gambar 2.1. Rangkaian pengganti dari motor DC [1]

Fungsi Transfer Sistem :

$$\frac{\dot{\theta}}{V} = \frac{K}{(Js + b)(Ls + R) + K^2} \dots\dots\dots(1)$$

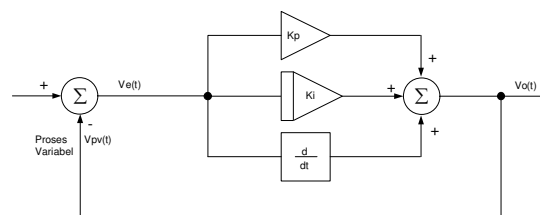
Keterangan :

- J : Momen Inersia dari rotor ( $kg.m^2/s^2$ )
- b : rasio redaman dari sistem mekanik ( $Nms$ )
- K : gaya elektromotif ( $Nm/Amp$ )
- R : resistansi (ohm)
- L : induktansi (H)
- V : tegangan input (Volt)
- $\dot{\theta}$  : kecepatan sudut (rps)

### 2.2 Sistem Kontrol PID

Sebuah kontrol PID adalah sebuah mekanisme kontrol lup umpan balik yang secara luas digunakan dalam sistem-sistem kontrol pada Industri. PID controller adalah solusi untuk memperbaiki *error* antara set point dan proses variabel dan outputnya merupakan hasil perbaikan yang dapat diatur dengan kriteria yang diinginkan.[2]

Algoritma dari PID kontroler terdiri dari 3 bagian seperti ditunjukkan pada gambar 2.2. Bagian tersebut adalah kontrol Proporsional, Integral dan Derivative. Masing-masing bagian memiliki konstanta. Dengan mengatur ketiga nilai konstanta dari kontrol PID maka kita bisa mendapatkan aksi kontrol sesuai dengan kriteria yang dibutuhkan. Sebuah catatan bahwa penggunaan kontrol PID tidak menjamin kontrol akan optimal dalam suatu sistem.[8]



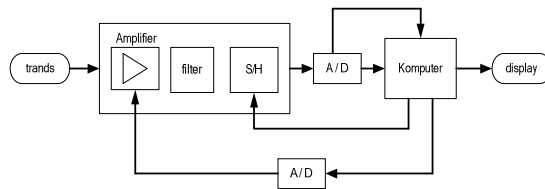
Gambar 2.2. Blok diagram PID.[6]

$$\begin{aligned}
 V_e(s) &= V_{sp}(s) - V_{out}(s) = V_{sp}(s) - V_p(s) \quad [5] \\
 V_{out}(s) &= V_e(s) \left( \frac{KdS^2 + KpS + Ki}{S} \right); \\
 &= (V_{sp}(s) - V_{out}(s)) \left( \frac{KdS^2 + KpS + Ki}{S} \right) \\
 &= \left( \frac{KdS^2 + KpS + Ki}{S} \right) V_{sp}(s) - \left( \frac{KdS^2 + KpS + Ki}{S} \right) V_{out}(s) \\
 V_{out}(s) \left( 1 + \left( \frac{KdS^2 + KpS + Ki}{S} \right) \right) &= \left( \frac{KdS^2 + KpS + Ki}{S} \right) V_{sp}(s) \\
 V_{out}(s) &= \left( \frac{\left( \frac{KdS^2 + KpS + Ki}{S} \right)}{\left( \frac{KdS^2 + (Kp + 1)S + Ki}{S} \right)} \right) V_{sp}(s) \\
 V_{out}(s) &= \left( \frac{KdS^2 + KpS + Ki}{KdS^2 + (Kp + 1)S + Ki} \right) V_{sp}(s) \dots\dots\dots(2)
 \end{aligned}$$

**2.3 Data Akuisisi**

Sistim akuisisi data dapat didefinisikan sebagai suatu sistem yang berfungsi untuk mengambil, mengumpulkan dan menyiapkan data, hingga memprosesnya untuk menghasilkan data yang dikehendaki. Jenis serta metode yang dipilih pada umumnya bertujuan untuk menyederhanakan setiap langkah yang dilaksanakan pada keseluruhan proses.

Sistem kanal tunggal yang disebut juga sistem akuisisi data sederhana, ditunjukan pada gambar 2.3.



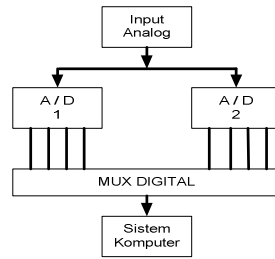
Gambar 2.3. Sistem akuisisi data kanal tunggal.[8]

Adapun fungsi masing-masing blok dalam sistem adalah sebagai berikut :

- Transduser : berfungsi untuk merubah besaran fisis yang diukur kedalam bentuk sinyal listrik.
- Amplifier : berfungsi untuk memperbesar amplitudo dari sinyal yang dihasilkan transduser.
- Filter : berfungsi untuk membatasi lebar band frekuensi sinyal listrik dari data yang diukur.
- S/H : berfungsi untuk menjaga amplitudo sinyal analog tetap konstan selama waktu konversi analog ke digital.
- A/D : berfungsi untuk merubah besaran analog kedalam bentuk representasi numerik.
- D/A : berfungsi untuk merubah besaran numerik kedalam sinyal analog.
- Komputer : berfungsi untuk mengolah data dan mengontrol proses.

Pada konfigurasi kanal tunggal, komputer berfungsi sebagai pemroses data dan juga pengontrol penguatan sinyal.

Sistem akuisisi data yang menggunakan komputer digital sebagai pengolah datanya, maka kecepatan ditentukan oleh proses pengubahan sinyal analog ke digital. Untuk mempercepat akuisisi data biasanya digunakan suatu konverter analog ke digital yang berkecepatan tinggi yang disebut dengan FLASH A to D. Bila kecepatan akuisisi masih ingin dipercepat, maka dapat digunakan teknik seperti yang diperlihatkan pada gambar 2.4. Cara ini digunakan dua buah A/D yang bekerja secara bergantian.



Gambar 2.4. Dua Buah A/D untuk Mempercepat Akuisisi

Satu diantara sekian banyak produk DAQ (data acquisition) yang memenuhi persyaratan tersebut adalah DAQ USB 6008 produk National Instruments seperti diperlihatkan pada gambar 2.5.

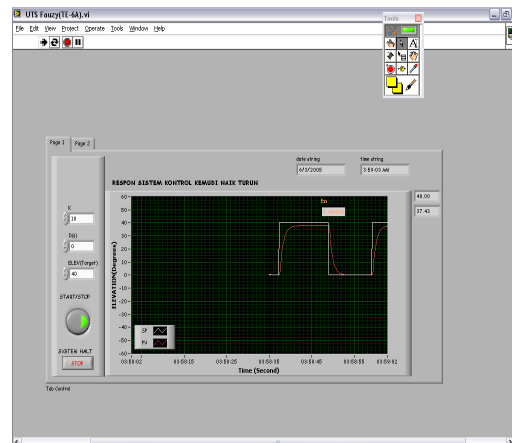


Gambar 2.5. Visualisasi NI-DAQ 6008 USB.[3]

**2.4 LabVIEW 2010**

LabVIEW (*Laboratory Virtual Instrumentation Engineering Workbench*) adalah media dan pengembangan berupa bahasa program *visual* dari National Instrument. Bahasa grafik dinamakan “G”. LabVIEW digunakan untuk akuisisi data, pengontrolan instrumen, dan otomatisasi industri pada variasi tampilan.[7]

Bahasa pemrograman LabVIEW adalah *dataflow programming language*. Pengeksekusian didefinisikan oleh struktur dari blok diagram yang mengkoneksikan berbagai macam fungsi dengan pengkabelan. Lalu tampilannya dapat dilihat pada *front panel* sebagai *Human Machine Interface* (HMI) seperti ditunjukkan pada gambar 2.6.



Gambar 2.6. Contoh Front Panel pada LabVIEW 2010

**2.5 PID LabVIEW**

PID controller membandingkan antara *setpoint* (SP) dengan *process variable* (PV) untuk mendapatkan nilai *error* (e).[4]

$$e = SP - PV \dots\dots\dots(3)$$

Kemudian PID controller menghitung aksi kontrol,  $u(t)$ , dimana  $K_c$  adalah *gain* controller.

$$u(t) = K_c \left( e + \frac{1}{T_i} \int_0^t e dt + T_d \frac{de}{dt} \right) \dots\dots\dots(4)$$

Jika *error* dan output controller memiliki range yang sama, -100% hingga 100%, *gain* controller berbanding terbalik dari proporsional band.  $T_i$  adalah waktu integral dalam menit, disebut juga waktu *reset* (*reset time*), dan  $T_d$  adalah waktu *derivative* dalam menit, disebut juga waktu rata-rata (*rate time*). Persamaan (5) adalah persamaan untuk menghitung aksi proporsional.

$$u_p(t) = K_c \cdot e \dots\dots\dots(5)$$

Persamaan (6) adalah persamaan untuk menghitung aksi *integral*.

$$u_I(t) = \frac{K_c}{T_i} \int_0^t e dt \dots\dots\dots(6)$$

Persamaan (7) adalah persamaan untuk menghitung aksi *derivative*.

$$u_D(t) = K_c T_d \frac{de}{dt} \dots\dots\dots(7)$$

Kemudian berikut ini adalah penjelasan implementasi algoritma PID LabVIEW 2010:

**Perhitungan Error**

Persamaan (8) menggambarkan arus *error* yang digunakan untuk menghitung proporsional, integral, dan aksi *derivative*.

$$e(k) = (SP - PV_f) \dots\dots\dots(8)$$

**Aksi Proporsional**

Aksi proporsional adalah perkalian dari *gain* controller dengan *error*, seperti diperlihatkan pada persamaan (2.9).

$$u_p(k) = (K_c * e(k)) \dots\dots\dots(9)$$

**Integral Trapezoid**

Integral trapezoid digunakan untuk menghindari perubahan ketajaman dalam aksi integral ketika terjadi perubahan secara tiba-tiba dalam *PV* atau *SP*. Penggunaan pengaturan nonlinier pada aksi integral untuk meniadakan *overshoot*. *Error* terbesar, aksi integral terkecil, seperti diperlihatkan pada persamaan (10).

$$u_I(k) = \frac{K_c}{T_i} \sum_{i=1}^k \left[ \frac{e(i) + e(i-1)}{2} \right] \Delta t \dots\dots\dots(10)$$

**Aksi Derivative Parsial**

Karena perubahan secara tiba-tiba pada *SP*, aksi *derivative* hanya berlaku pada *PV*, bukan pada *error* (*e*), untuk menghindari sentakan *derivative*. Persamaan (11) menggambarkan aksi *derivative* parsial.

$$u_D(k) = -K_c \frac{T_d}{\Delta t} (PV_f(k) - PV_f(k-1)) \dots\dots\dots(11)$$

**Output Controller**

Output controller adalah penyajian terakhir dari proporsional, integral dan aksi *derivative*, sebagaimana diperlihatkan pada persamaan (12).

$$u(k) = u_p(k) + u_I(k) + u_D(k) \dots\dots\dots(12)$$

**Pembatasan Output**

Output controller sesungguhnya dibatasi pada range yang ditetapkan untuk output kontrol.

$$\begin{aligned} & \text{If } u(k) \geq u_{\max} \text{ then } u(k) = u_{\max} \\ & \text{dan} \\ & \text{If } u(k) \leq u_{\min} \text{ then } u(k) = u_{\min} \end{aligned}$$

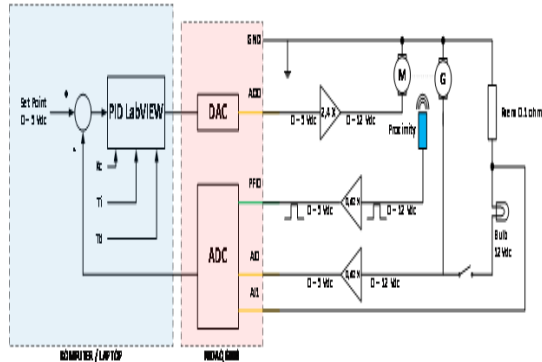
Persamaan (13) memperlihatkan model praktis dari PID controller.

$$u(t) = K_c \left[ (SP - PV) + \frac{1}{T_i} \int_0^t (SP - PV) dt - T_d \frac{dPV_f}{dt} \right] \dots\dots\dots(13)$$

**3. HASIL DAN PEMBAHASAN**

**3.1 Perancangan Sistem**

- Nama Sistem : Kontrol Kecepatan Motor DC dengan PID LabVIEW 2010
- Fungsi Sistem : Mengontrol kecepatan motor DC agar tidak ada *overshoot* dan *error steady state* yang terjadi 2 – 5%. Selain itu menjaga kecepatan motor tetap stabil meskipun terdapat gangguan dari luar sistem.
- Diagram Blok :



Gambar 3.1. Diagram Blok system

Berikut ini adalah keterangan bagian dari system, yaitu:

- Komputer/Laptop sebagai pengekseskusi Program LabVIEW 2010;
- NIDAQ 6008 sebagai Sistem Akuisisi Data;
- Amplifier sebagai penguat tegangan output dari NIDAQ 6008 dengan gain 2,4 kali;
- Attenuator sebagai pelemah tegangan output dari proximity dan generator sehingga dapat dibaca oleh NIDAQ 6008, gain pelemahan adalah 0,42 kali;
- Motor sebagai Objek yang akan dikontrol;
- Generator sebagai penghasil tegangan DC akan digunakan untuk memonitor RPM dan PV;
- Proximity Sensor sebagai Acuan Kalibrasi RPM motor.

Sistem ini memiliki cara kerja sebagai berikut yaitu *set point* ditentukan oleh operator melalui *front panel*. Kemudian PID LabVIEW akan memproses error dari set point dan process variable (umpan balik dari generator DC). Kemudian memberikan outputnya ke motor DC. Jika generator dc diberikan beban resistif dengan mengaktifkan saklar, maka PID LabVIEW akan mendeteksi perubahan dari error antara setpoint dan process variable. Kemudian PID LabVIEW akan secara otomatis memperkecil error yang terjadi sehingga sistem akan kembali mengejar nilai set point.

Kontroller PID ini dirancang dengan nilai  $K_c=0,01$ ,  $T_i=0,001$  menit = 0,06 (detik) dan  $T_d=1.10^{-4}$  menit ( $6 \cdot 10^{-3}$  detik). Berdasarkan persamaan berikut:

$$u(t) = K_c \left( e + \frac{1}{T_i} \int_0^t e dt + T_d \frac{de}{dt} \right)$$

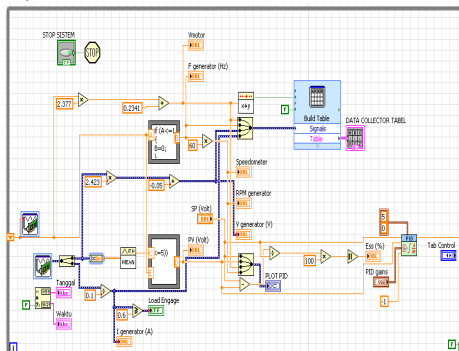
$$U(s) = K_c \left( 1 + \frac{1}{T_i s} + T_d s \right) \times E(s)$$

$$U(s) = 0,01 \left( 1 + \frac{1}{0,06s} + 0,006s \right) \times E(s)$$

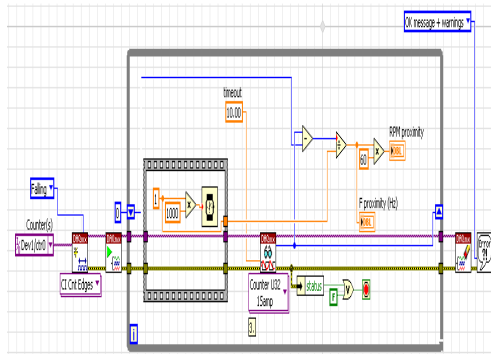
Jadi, fungsi transfer lup terbukanya adalah

$$U(s) = \left( \frac{0,00036s^2 + 0,06s + 1}{s} \right) E(s) \dots\dots\dots(14)$$

Perancangan sistem kontrol kecepatan motor DC dengan PID LabVIEW 2010 dilakukan dengan cara membuat software kontrol lup tertutup seperti blok diagram kontrol PID pada gambar 3.1. Hasil perancangan diperlihatkan pada gambar 3.2, 3.3, 3.4 dan 3.5.



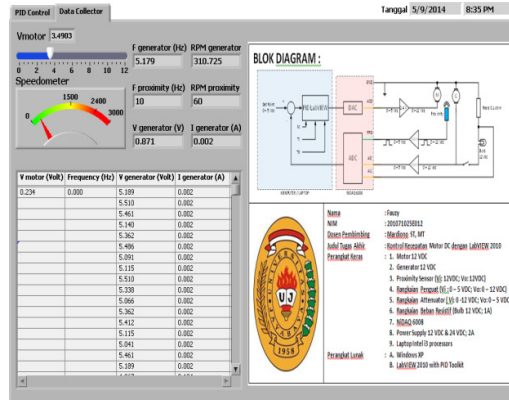
Gambar 3.2. Blok Diagram Kontrol Kecepatan Motor DC dengan PID LabVIEW 2010



Gambar 3. 3 Blok Diagram Penghitung Frekuensi Motor DC



Gambar 3.4. PID Front panel Kontrol Kecepatan Motor DC dengan PID LabVIEW 2010

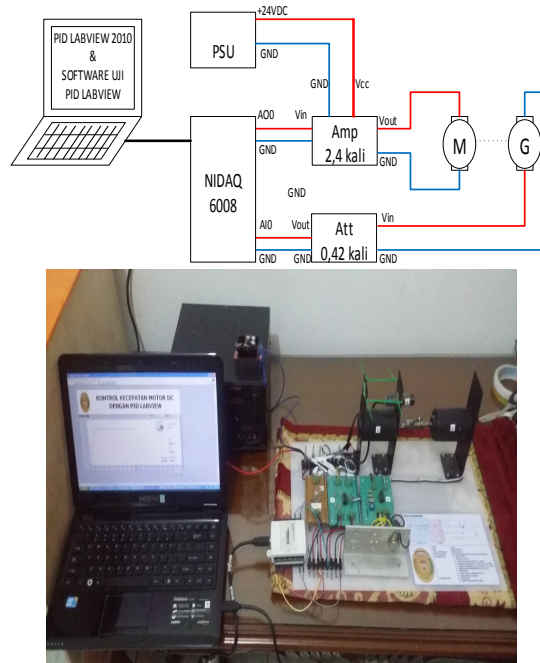


Gambar 3.5. Data Collector Front panel Kontrol Kecepatan Motor DC dengan PID LabVIEW 2010

3.2 Hasil Pengukuran

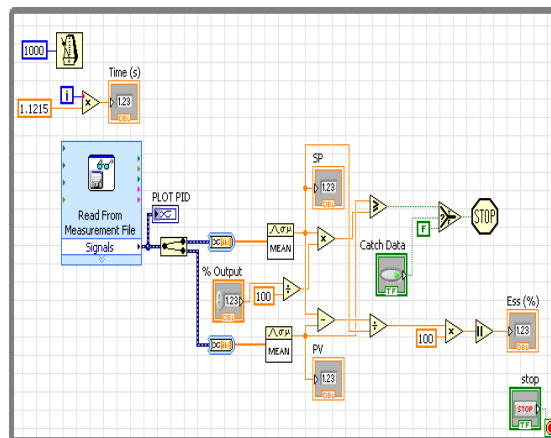
Pengujian dimaksudkan untuk mendapatkan data-data yang diperlukan untuk kemudian dianalisis, agar kinerja dan spesifikasi sistem setelah direalisasi dapat diketahui (*settling-time, delay-time, overshoot, steady state error* dan kemampuan melakukan koreksi terhadap pembebanan). Penyimpangan nilai parameter di luar batas yang dipersyaratkan akan digunakan sebagai masukan untuk melakukan perbaikan. Sampel data numerik hasil pengukuran secara otomatis disimpan dalam satu file (text file) dengan ekstensi \*.lvm. Untuk keperluan analisis, data ini dapat di-replay dan diolah sesuai kebutuhan.

Konfigurasi untuk pengujian sistem kontrol kecepatan motor DC dengan PID LabVIEW 2010 diperlihatkan pada gambar 3.6.



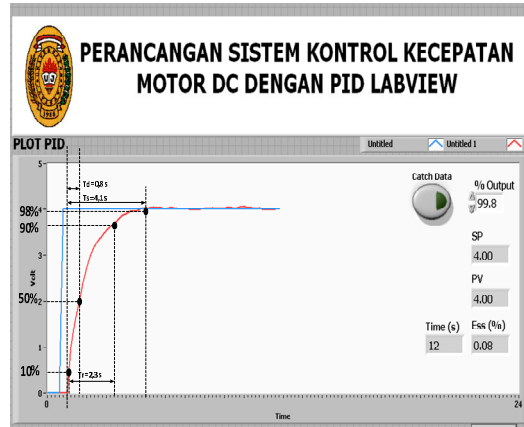
Gambar 3.6. Konfigurasi Pengujian sistem kontrol kecepatan motor DC dengan PID LabVIEW 2010

Dengan membuat program pembacaan data seperti diperlihatkan pada gambar 4.3, kita dapat mengambil kembali data yang telah disimpan pada saat pengujian untuk selanjutnya diolah agar dapat diketahui spesifikasi dari sistem kontrol yang digunakan. Analisa data dilakukan secara otomatis dengan membuat program untuk menganalisa respon dari sistem. Hasil analisa diperlihatkan pada gambar 4.4.



Gambar 3.7. Program pembacaan data hasil pengukuran





Gambar 3.8. Pengukuran settling time (ts), rise time (tr), dan delay time (td)

Dari data yang diperoleh maka diketahui bahwa sistem kontrol memiliki spesifikasi sebagai berikut:

- *Settling time*,  $t_s = 4,1 \text{ s}$
- *Rise time*,  $t_r = 2,3 \text{ s}$
- $E_{ss} (\%) = 1 \%$  (dibawah 5%)
- *Overshoot*,  $M_p = 0 \%$
- *Delay-time*,  $t_d = 0,8 \text{ s}$

### 3.3 Pembahasan Hasil Pengukuran

Perhitungan secara teoritis, fungsi transfer lup terbuka dari PID, yaitu:

$$U(s) = \left( \frac{0.00036s^2 + 0.06s + 1}{s} \right) E(s)$$

Fungsi transfer lup tertutup dari PID unity feedback:

$$U(s) = \left( \frac{s^2 + 166,67s + 2777,78}{s^2 + 2944,44s + 2777,78} \right) V_{in}(s)$$

Saat  $V_{in} = 4\text{Volt}$  ; unit step

$$U(s) = \left( \frac{s^2 + 166,67s + 2777,78}{(s + 0,944)(s + 2943,5)} \right) \frac{4}{s}$$

Setelah diproses secara matematis adalah :

$$U(s) = \frac{k1}{s} + \frac{k2}{(s + 0,944)} + \frac{k3}{(s + 2943,5)}$$

Dalam hal ini  $k3$  tidak perlu dicari nilainya karena nilai eksponensial pada  $U(t)$  akan bernilai mendekati nol.

$$k1 = \left[ U(s)s \right]_{s=0} = 4$$

$$k2 = \left[ U(s)(s + 0,944) \right]_{s=-0,944} = -4,23$$

Jadi,

$$U(t) = 4 - 4,23 e^{-0,944t} \dots\dots\dots(4.1)$$

Untuk setting time yaitu saat output 98 %, masukkan ke dalam persamaan 4.1 dan carilah nilai t, yaitu :

$$t_s = 4,2 \text{ detik} \dots\dots\dots(4.2)$$

Untuk waktu naik yaitu saat output antara 10 % sampai 90 %, masukkan ke dalam persamaan 4.1 dan carilah nilai t, yaitu :

Untuk nilai output 10 % maka  $t_{(10\%)} = 0,17 \text{ detik}$

Untuk nilai output 90 % maka  $t_{(90\%)} = 2,50 \text{ detik}$

Jadi,

$$t_r = t_{(90\%)} - t_{(10\%)} = 2,33 \text{ detik} \dots\dots\dots(4.3)$$

tidak ada *overshoot* secara teoritis

Membandingkan data hasil perhitungan dengan pengujian, data hasil perhitungan :

- *Settling time*,  $t_s = 4,2 \text{ detik}$
- *Rise time*,  $t_r = 2,33 \text{ detik}$
- $E_{ss} (\%) = 1 \%$  (dibawah 5%)

➤ *Overshoot*,  $M_p = 0\%$  (dibawah 5%)

➤ *Delay-time*,  $t_d = 0,8$  detik

Data hasil pengujian :

➤ *Settling time*,  $t_s = 4.1$  detik

➤ *Rise time*,  $t_r = 2,3$  detik

➤  $Ess (\%) = 0,82\%$  (dibawah 5%)

➤ *Overshoot*,  $M_p = 0\%$  (dibawah 5%)

➤ *Delay-time*,  $t_d = 0,8$

$$\% \text{ penyimpangan} = \frac{\text{data perhitungan} - \text{data pengujian}}{\text{data perhitungan}} \times 100\%$$

Untuk *settling time* persentase perbedaannya adalah 2,3 %

Untuk *delay time* persentase perbedaannya adalah 0 %

Untuk *rise time* persentase perbedaannya adalah 1,3 %

#### 4. SIMPULAN

Berdasarkan hasil pengamatan dan analisis data hasil pengujian, dapat disimpulkan bahwa sistem kontrol kecepatan dan pengereman menggunakan kontrol PID *LabVIEW 2010* memiliki nilai *overshoot sama dengan nol*. Kemudian penyimpangan yang terjadi antara pengujian dan perhitungan adalah 2, 3% untuk *settling time* dan 1,3% untuk *rise time* dan 0% untuk *delay time*.

#### DAFTAR PUSTAKA

- [1]. Carnegie Mellon, *Modeling DC Motor Position*, The University of Michigan, 2001.
- [2]. Carnegie Mellon, *Tutorial PID*, The University of Michigan, 1817.
- [3]. National Instruments, *Datasheet USB-6008/6009 User Guide and Specification*, 2006.
- [4]. National Instruments, *PID and Fuzzy Control Toolset User Manual*, USA: Austin Texas, 2006.
- [5]. Philips dan Harbor, *Feedback Control Systems*, London: Prentice-Hall International Editions, 1991.
- [6]. Endang Wijaya, *Tutorial Sistem Kontrol Analog*, Depok: Politeknik Negeri Jakarta, 2006.
- [7]. LabVIEW 8.0: *Measurement and Automation*, National Instruments, USA: Austin Texas, 2006
- [8] Jacob J.Michael, *Industrial control Electronic*, New Jersey: Prentice Hall of Englewood cliffs.