# PEMBUATAN PERANGKAT LUNAK MENGGUNAKAN MATLAB/SIMULINK UNTUK PENGENDALIAN KECEPATAN MOTOR DC DENGAN METODE KONVENSIONAL

Hasil Penelitian / Pemikiran yang tidak dipublikasikan
Disusun sebagai salah satu syarat untuk
Kenaikan Angka Kredit Jabatan Fungsional Lektor

Oleh
Pipit Anggraeni
197908242005012001

JURUSAN TEKNIK OTOMASI MANUFAKTUR
DAN MEKATRONIKA
POLITEKNIK MANUFAKTUR NEGERI BANDUNG BANDUNG
2009

# Pembuatan Perangkat Lunak Menggunakan MATLAB/Simulink untuk Pengendalian Kecepatan Motor DC dengan Metode Konvensional

**Pipit Anggraeni**
Jurusan Teknik Otomasi Manufaktur dan Mekatronika – POLMAN Bandung, Jl. Kanayakan 21 Bandung,
Email : pipit_anggraeni@polman-bandung.ac.id

**Abstrak**

*Pada penelitian ini dibahas mengenai aplikasi MATLAB / Simulink untuk kendali kecepatan motor DC dengan metode konvensional (P, I, D). Simulasi akan menunjukkan pengaruh perubahan nilai parameter kendali terhadap pengaruh respon sistem. Metode Ziegler-Nichols digunakan untuk penalaan parameter kendali. Mikrokontroler AVR ATMega8535 digunakan sebagai interface yang menghubungkan PC dengan perangkat keras. Respon sistem ditampilkan dalam bentuk kurva . Hasil pengujian menunjukkan bahwa dengan metoda konvensional, sistem memberikan respon yang baik, dilihat dari setting time dan steady state yang lebih baik dari respon sistem tanpa pengendali.*

**Abstract**

*This paper describes about application of MATLAB/Simulink for controlling the speed of DC motor by using conventional method (P, I, D). Simulations will show the effect of variable parameter value in system respond. Ziegler-Nichols method is used in tuning parameter value. Microcontroller AVR ATMega8535 is used as interface which connecting PC with hardware. System responds are shown in curve form. The result of trials show that system gives good respond by the implementation of conventional method. It can be shown from better value of setting time and steady state if it is compared with the condition when there is no controller in system.*

## 1. Pendahuluan

Beberapa tahun terakhir, mulai dikembangkan simulasi sistem kendali dengan menggunakan perangkat lunak MATLAB/Simulink. Perangkat lunak tersebut mempunyai banyak sumber daya yang memungkinkan berbagai simulasi pemodelan, visualisasi, dan model *plant* kendali.

Sistem kendali yang baik harus mempunyai respon yang cepat dan akurat serta tahan terhadap *disturbance*. Pada metode kendali konvensional, MATLAB/Simulink dapat membantu perancang untuk melihat respon berbagai kombinasi parameter kendali dengan variasi masukan. Sehingga didapat parameter kendali yang sesuai untuk sebuah sistem kendali.

Pada penelitian ini dibahas aplikasi MATLAB / Simulink untuk kendali kecepatan motor DC dengan metode konvensional (P, I, D). Simulasi akan menunjukkan pengaruh perubahan nilai parameter kendali terhadap pengaruh respon sistem.

Tujuan umum yang ingin dicapai dari proyek akhir ini adalah mengendalikan kecepatan motor DC berbasis perangkat lunak MATLAB/Simulink pada PC. Sedangkan tujuan khusus yang ingin dicapai yaitu mengendalikan kecepatan motor DC dengan metode kendali konvensional, membuat simulasi pengendalian kecepatan motor DC dengan perangkat lunak MATLAB/Simulink, dan menyajikan perbandingan hasil simulasi yang dikirim ke motor DC dengan kondisi yang terjadi pada *plant* .

## 2. Landasan Teori

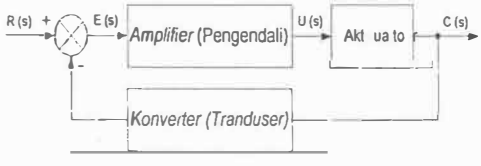
### 2.1 Pengendali

Unit pengendali adalah perangkat yang melakukan pengelolaan data untuk mengubah besaran masukan menjadi besaran keluaran yang diinginkan.

Jenis sistem kendali berdasarkan metoda regulasinya, yaitu :

1. Sistem kendali loop terbuka (*open loop control system*)
2. Sistem kendali loop tertutup (*close loop control system*)

Pada penelitian ini ditekankan pada sistem kendali loop tertutup.




**Gambar 2.1** *Sistem loop tertutup*

Unit pengendali harus terbentuk minimal tiga elemen dasar, yaitu :

1. Pembanding
2. Pengendali
3. *Converter/tranduser*.

Keluaran dari pembanding adalah perbedaan nilai yang terdapat antara besaran acuan (nilai yang diinginkan) dengan besaran keluaran yang dihasilkan. Perbedaan nilai ini bisa disebut sebagai *error signal* (e), e = besaran masukan – besaran keluaran [1].

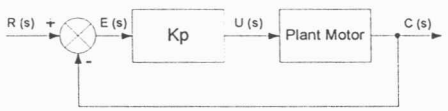
## 2.2 Jenis-Jenis Pengendali

### 2.2.1 Kendali proporsional (P)

Kendali proporsional memiliki keluaran sebanding dengan besarnya kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara sederhana, keluaran kendali proporsional adalah perkalian Kp dengan masukannya. Hubungan antara sinyal kendali u(t) dan sinyal *error* e(t), bila dinyatakan dalam besaran tranformasi Laplace adalah :

$$\frac{U(s)}{E(s)} = Kp \qquad (2.1)$$

Kp adalah konstanta proporsional.




**Gambar 2.2** *Blok diagram proporsional*

### 2.2.2 Kendali integral (I)

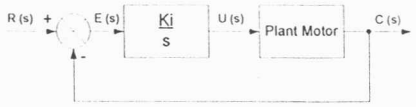
Kendali integral berfungsi menghasilkan respon sistem yang memiliki *error steady states = 0*. Keluaran pengendali dipengaruhi oleh perubahan yang sebanding dengan nilai *error*. Keluaran pengendali ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau tidak ada perubahan sinyal *error*, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan (sinyal kendali yang dihasilkan adalah konstan, dalam hal ini merupakan *setpoint*-nya sendiri). Hal ini disebabkan karena hasil integral dari angka nol adalah konstan, dalam bentuk rumus berikut :

$$\frac{U(s)}{E(s)} = \frac{Ki}{s} \qquad (2.2)$$

$$u(k) = Ki \int_0^t e(k)dt \qquad (2.3)$$

$$u(k) = Ki \sum_{j=0}^{k} e(j)\Delta t = u(k-1) + Ki.\Delta t.e(k)$$

Ki adalah konstanta waktu integral



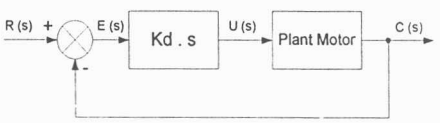

**Gambar 2.3** *Blok diagram integral*

### 2.2.3 Kendali diferensial (D)

Sifat pengendali diferensial adalah dalam konteks 'kecepatan' atau *rate* dari *error*. Umpan balik sebanding dengan kecepatan perubahan *e(t)* sehingga pengendali diferensial dapat mengantisipasi *error* yang akan terjadi. Perubahan mendadak pada masukan mengakibatkan perubahan yang sangat besar dan cepat. Pengendali diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja pengendali diferensial hanya efektif pada lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu pengendali diferensial tidak pernah digunakan tanpa ada pengendali lain dalam sebuah sistem. Sinyal kendali yang dihasilkan adalah sebagai berikut :

$$\frac{U(s)}{E(s)} = Kd.s \qquad (2.5)$$

$$u(k) = Kd\,[e(k) - e(k-1)]/\Delta t$$
$$= (Kd/\Delta t)[e(k) - e(k-1)] \qquad (2.6)$$




**Gambar 2.4** *Blok diagram derivative*

### 2.2.4 Kendali proporsional – integral (PI)

Pengendali PI merupakan gabungan dari pengendali proporsional dan integral. Aksi kendali PI didefinisikan dengan persamaan berikut :

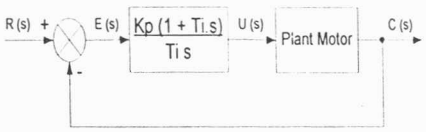
$$u(k) = Kp\,e(k) + \frac{Kp}{Ti}\int_0^t e(k)\,dt \qquad (2.7)$$

$$Ki = \frac{Kp}{Ti} \qquad (2.8)$$

Atau fungsi alih pengendali adalah

$$\frac{U(s)}{E(s)} = Kp\left(1 + \frac{1}{Tis}\right) \qquad (2.9)$$

Kp merupakan konstanta proporsional atau penguat, dan Ti menyatakan waktu integral yang mengatur aksi kontrol integral. Kp dan Ti dapat di atur.




**Gambar 2.5** *Blok diagram PI*

### 2.2.5 Kendali proporsional-derivatif (PD)

Aksi kontrol pengendali proporsional derivatif didefinisikan dengan rumus berikut:

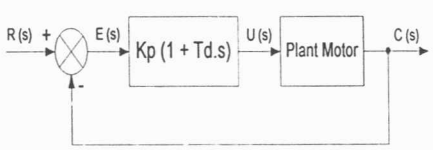
$$u(k) = Kp \,.\, e(k) + Kp \,.\, Td \,\frac{de(t)}{dt} \qquad (2.10)$$

$$Kd = Kp \,.\, Kd \qquad (2.11)$$

Atau fungsi alih pengendali adalah

$$\frac{U(s)}{E(s)} = Kp(1 + Td \,.\, s) \qquad (2.12)$$

Kp adalah konstanta proporsional. Td menyatakan waktu derivatif, yaitu selang waktu bertambah majunya respon aksi kontrol proporsional karena aksi laju. Kp dan Td dapat di atur.




**Gambar 2.6** *Blok diagram PD*

### 2.2.6 Kendali proporsional-integral-derivatif (PID)

Kendali PID merupakan gabungan dari kendali proporsional, integral, dan derivatif. Aksi kendali dari pengendali ini didefinisikan sebagai berikut : [1][2]

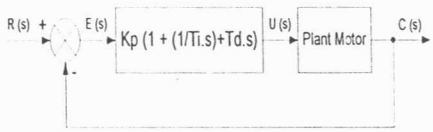
$$u(k) = Kp\,e(k) + Ki\int_0^t e(k)dt + Kd\frac{de}{dt}$$

$$u(k) = Kp\,e(t) + Ki\sum_{j=0}^{k} e(j)\Delta t + Kd\left[e(k) - e(k-1)\right]/\Delta t \qquad (2.9)$$

(2.14)

Atau fungsi alih pengendali adalah

$$\frac{U(s)}{E(s)} = Kp\left(1 + \frac{1}{Ti.s} + Td.s\right) \qquad (2.15)$$

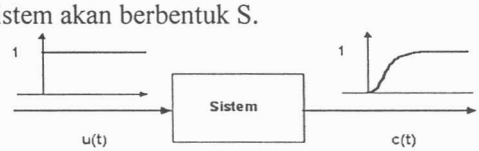



**Gambar 2.7** *Blok diagram PID*

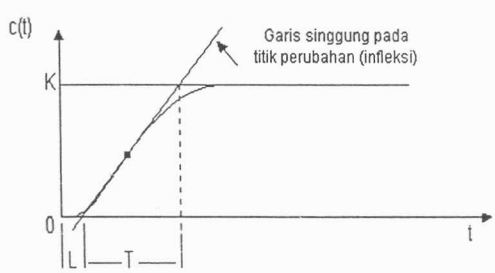
## 2.3 Penalaan PID dengan Metode Ziegler-Nichols

Metode Ziegler-Nichols merupakan suatu metode pendekatan eksperimental penalaan kontroller PID. Karena penyusunan model matematik *plant* tidak mudah, maka dikembangkan suatu metode eksperimental. Metode ini didasarkan pada reaksi *plant* yang dikenai suatu perubahan.

Metode Ziegler-Nichols pertama kali diperkenalkan pada tahun 1942. Metode ini memiliki dua cara, kurva reaksi dan metode osilasi. Kedua metode ditujukan untuk menghasilkan respon sistem dengan lonjakan maksimum 25%. Metode kurva reaksi berdasar reaksi sistem *loop* terbuka. *Plant* sebagai *loop* terbuka diberi masukan sinyal fungsi *step,* bila *plant* minimal tidak mengandung unsur integrator atau pole-pole kompleks, reaksi sistem akan berbentuk S.




**Gambar 2.8** *Respon tangga satuan system*




**Gambar 2.9** *Kurva respon berbentuk S*

Kurva berbentuk S mempunyai dua konstanta, waktu mati (*dead time*) L dan waktu tunda T. Dari gambar 2.15 terlihat bahwa kurva reaksi berubah naik, setelah selang waktu L. Sedangkan waktu tunda

menggambarkan perubahan kurva setelah mencapai 66% dari keadaan mantap. Pada kurva dibuat suatu garis yang bersinggungan dengan garis kurva. Garis singgung memotong sumbu absis dan garis maksimum. Perpotongan garis singgung dengan sumbu absis merupakan ukuran waktu mati, dan perpotongan dengan garis maksimum merupakan waktu tunda yang diukur dari titik waktu L. Penalaan parameter PID berdasarkan perolehan kedua konstanta itu. Tabel 2.1 merupakan rumusan penalaan parameter PID berdasarkan cara kurva reaksi. [6]

*Tabel 2.1 Penalaan paramater PID dengan metode kurva reaksi*

| Tipe Kontroler | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | T/L | ~ | 0 |
| PI | 0,9 T/L | L/0.3 | 0 |
| PID | 1,2 T/L | 2 *L* | 0,5 *L* |

## 2.4 MATLAB/Simulink

MATLAB merupakan perangkat lunak pemrograman perhitungan dan analisis yang banyak digunakan dalam semua area penerapan matematika, pada bidang pendidikan, penelitian di universitas, dan industri. MATLAB merupakan singkatan dari MATriks LABoratory, artinya perangkat lunak ini dibuat berdasarkan vektor dan matrik. Perangkat lunak ini awalnya banyak digunakan pada studi aljabar linier, serta merupakan perangkat yang tepat untuk menyelesaikan persamaan aljabar dan diferensial dan juga untuk integrasi numerik. Dalam hal pemrograman, MATLAB serupa dengan bahasa C dan bahkan salah satu bahasa pemrograman termudah dalam penulisan program matematik.

MATLAB dapat membuat program dan memiliki fitur lain yang memungkinkan MATLAB digunakan sebagai *tools* yang memudahkan disain dan analisis matematis. MATLAB berorientasi untuk memudahkan penuangan rumus perhitungan matematis, memungkinkan pembuatan program matematis yang kompleks bisa menjadi lebih mudah, namun bisa jadi eksekusi program MATLAB ini jauh lebih lambat dibandingkan bila dibuat dengan perangkat pemrograman lainnya.

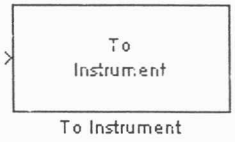
Dewasa ini dibutuhkan program komputer yang menyediakan *tools* komputasi, pemodelan, dan simulasi dengan berbagai fasilitasnya. Maka, MATLAB dilengkapi dengan berbagai fitur , yaitu Simulink, *Toolbox,* GUIDE dan lain-lain. Hasil program MATLAB dapat diekspor ke C/C++, Visual Basic, Fortran, COM, Java, dan web/internet. Hasil dari MATLAB dapat dikompilasi dan menjadi program yang lebih cepat dieksekusi, serta bisa diakses dengan berbagai cara. [6]

Simulink adalah bagian dari MATLAB. Simulink adalah sebuah perangkat lunak yang digunakan untuk pemodelan, simulasi, dan analisis sistem dinamis. Simulink dapat mengolah sistem linier maupun non linier, model dalam waktu kontinyu, *sample*, atau gabungan dari keduanya. Simulink menyediakan GUI *(Graphical User Interface)* untuk pemodelan sistem sebagai diagram blok.

Fitur Simulink lebih dikenal dengan nama *toolbox*. *Toolbox* adalah kumpulan dari fungsi MATLAB (M-files) yang telah dikembangkan ke suatu lingkungan kerja MATLAB untuk memecahkan masalah. Beberapa area yang sudah terpecahkan dengan *toolbox* meliputi pengolahan sinyal, sistem kontrol, *neural networks*, *fuzzy logic*, *real-time workshop* , dan lain-lain. [7]

## 2.5 Instrument Control Toolbox 2.8

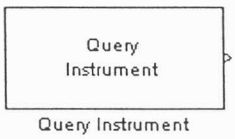
*Instrument Control Toolbox 2.8* adalah *toolbox* yang digunakan untuk komunikasi serial antara MATLAB/Simulink (PC) dengan mikrokontroler. *Toolbox* ini menyediakan fasilitas *To Instrument* untuk mengirim data dari Simulink ke instrumen (mikrokontroler), serta fasilitas *Query Instrument* menerima data dari instrumen untuk kemudian dianalisis dan ditampilkan di Simulink.[8]




**Gambar 2.10** *Blok To Instrument*

Blok To Instrument digunakan untuk mengirim data dari Simulink (PC) ke instrumen (mikrokontroler). Blok ini mengirim data ke instrumen selama mode *run* dalam Simulink diaktifkan. Berikut adalah blok parameter pengaturan mode pengiriman serta jenis data yang dikirim. *Interface* yang dipilih adalah komunikasi serial, dengan *baudrate* 9600. Tipe data yang dikirim adalah ASCII

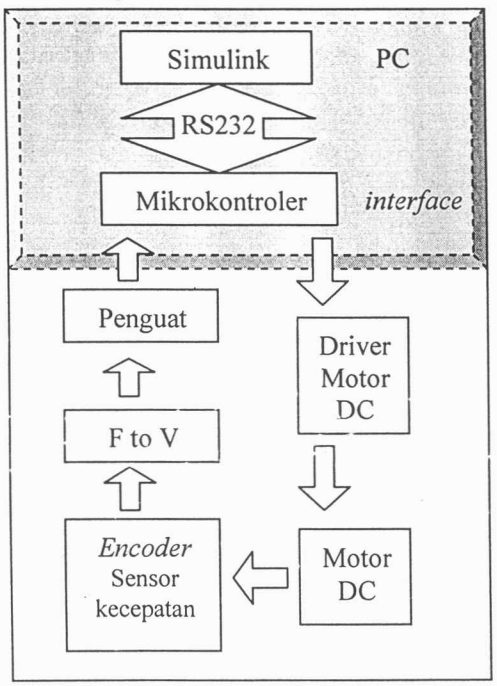
dengan format string '%d' yaitu *signed decimal integer.*




**Gambar 2.11** *Blok Query Instrument*

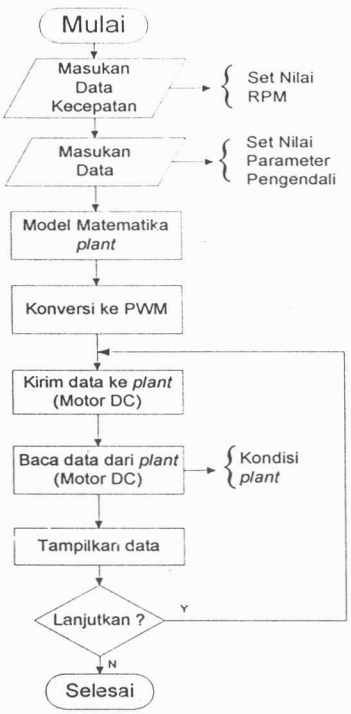
Blok *Query Instrument* digunakan oleh Simulink (PC) untuk menerima data dari instrumen (mikrokontroler). Blok ini menerima data selama mode *run* dalam Simulink diaktifkan. Untuk memastikan bahwa data yang diterima adalah benar, maka *Query Instrument* hanya menerima data setelah ada data yang dikirim ke instrumen melalui blok *To Instrument*. Berikut adalah blok parameter pengaturan mode terima serta jenis data yang diterima. *Interface* yang dipilih adalah komunikasi serial, dengan *baudrate* 9600. Tipe data yang diterima adalah biner dengan format 8 bit *unsigned* integer, sehingga bit ke-8 dibaca sebagai data MSB, bukan dibaca sebagai penanda bilangan positif atau negatif.

## 3. Perancangan Dan Pembuatan Sistem




**Gambar 3.1** *Blok sistem*

Identifikasikan masalah dibagi menjadi tiga bagian. Bagian pertama yaitu aplikasi Simulink untuk pengendalian motor DC dengan metode konvensional. Bagian kedua adalah komunikasi antara Simulink (PC) dengan motor DC. Bagian terakhir yaitu penggunaan mikrokontroler AVR ATMega8535 sebagai *interface* antara PC dengan *hardware* .




**Gambar 3.2** *Flowchart sistem*

## 4. Uji Coba Dan Hasil

**Tabel 4.1** Data Hasil Pengujian Penalaan PID

| No | V out (V) | Waktu mati L (ms) | Waktu tunda T (ms) |
|---|---|---|---|
| 1 | 10.8 | 20 | 130 |
| 2 | 10.8 | 20 | 140 |
| 3 | 10.8 | 20 | 120 |
| 4 | 10.8 | 10 | 120 |
| 5 | 10.8 | 20 | 110 |
| 6 | 10.8 | 20 | 160 |
| 7 | 10.8 | 20 | 160 |
| R | 10.8 | 18.57 | 134.29 |

Berdasar metoda Ziegler-Nichols, perhitungan nilai parameter kendali dengan menggunakan data hasil pengujian *tuning loop* terbuka menghasilkan nilai berikut :

Kp= 1.2 . [(rerata T) / (rerata L)]

= 1.2 . (134.29 / 18.57)

*Kp = 8.68*

Ti = 2 . rerata L

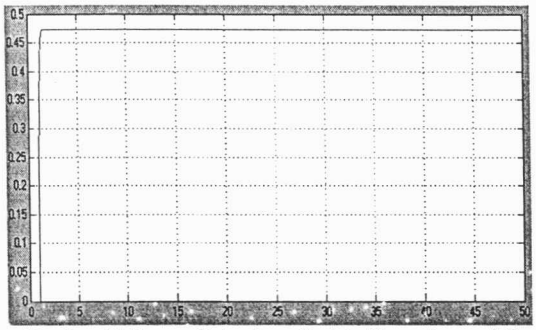
= 2 . 18.57 ms

*Ki = 26.93*

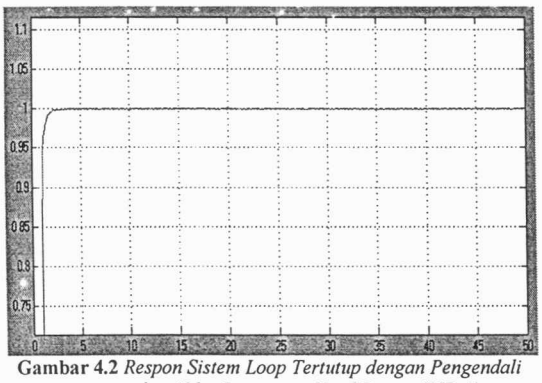
Td = 0.5 . rerata L

= 0.5 . 18.57 ms

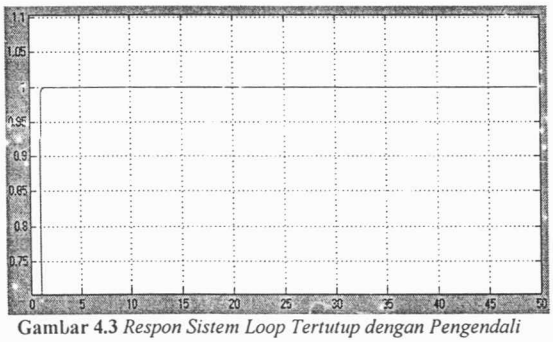
*Kd = 0.0093*



**Gambar 4.1** *Respon Sistem Loop TertutupTanpa Pengendali*



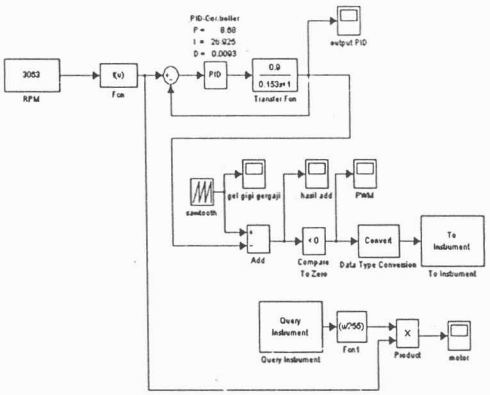
**Gambar 4.2** *Respon Sistem Loop Tertutup dengan Pengendali menggunakan Nilai Parameter Hasil Tuning(PID 1)*



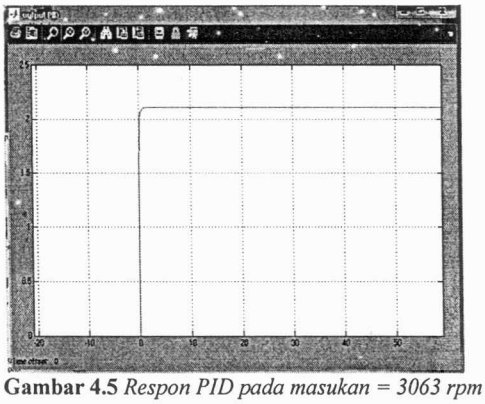
**Gambar 4.3** *Respon Sistem Loop Tertutup dengan Pengendali menggunakan Pengubahan Nilai Parameter Hasil Tuning(PID2)*

**Tabel 4.2** Perbandingan respon sistem

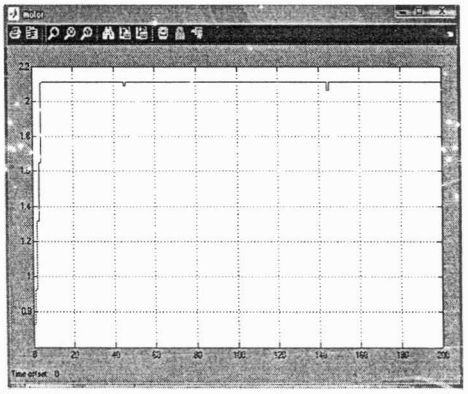
| | Tanpa Pengendali | PID 1 | PID 2 |
|---|---|---|---|
| Kp | 0 | 8.68 | 8.68 |
| Ki | 0 | 26.9 | 50 |
| Kd | 0 | 0.0093 | 0.0093 |
| | | | |
| Tr (s) | 1.45 | 2.67 | 1.392 |
| *Out* (V) | 0.48 | 1 | 1 |



**Gambar 4.4** *Model Sistem Kendali*



**Gambar 4.5** *Respon PID pada masukan = 3063 rpm*

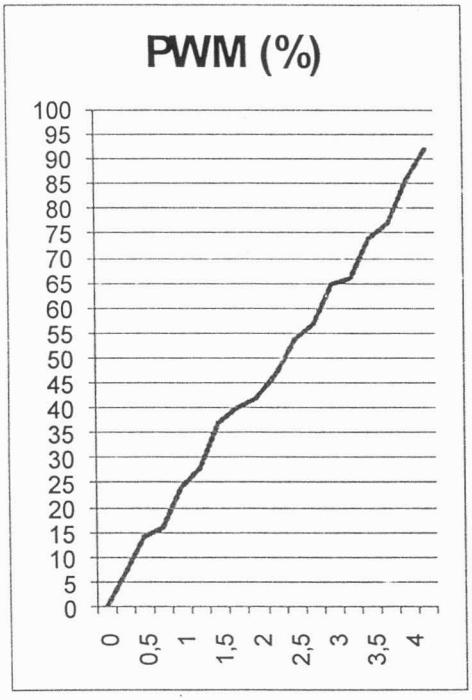


**Gambar 4.6** *Respon motor pada masukan = 3063 rpm*

**Tabel 4.3**. *Data hasil pengujian PWM*

| *INPUT* ( V ) | *DUTY CYCLE* ( % ) | | *INPUT* ( V ) | *DUTY CYCLE* ( % ) |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 0.00 | 0 | | 47 | 47 |
| 0.25 | 7 | | 54 | 54 |
| 0.50 | 14 | | 57 | 57 |
| 0.75 | 16 | | 3.00 | 65 |
| 1.00 | 24 | | 3.25 | 66 |
| 1.25 | 28 | | 3.50 | 74 |
| 1.50 | 37 | | 3.75 | 77 |
| 1.75 | 40 | | 4.00 | 86 |
| 2.00 | 42 | | 4.50 | 92 |

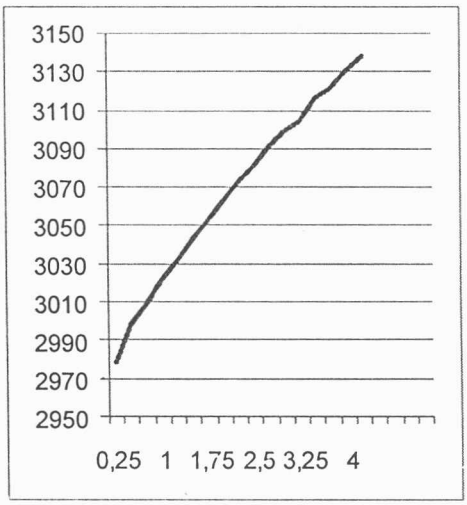



**Gambar 4.7**. *Kurva pengaturan input terhadap duty cycle PWM*

Sumbu X : *input* ( V )
Sumbu Y : *duty cycle* PWM ( % )

**Tabel 4.4**. *Data hasil pengujian kecepatan*

| *INPUT* ( V ) | Kecepatan ( rpm ) | | *INPUT* ( V ) | Kecepatan ( rpm ) |
|---|---|---|---|---|
| 0.00 | 0 | | 47 | 3073 |
| 0.25 | 2979 | | 54 | 3081 |
| 0.50 | 2998 | | 57 | 3091 |
| 0.75 | 3008 | | 3.00 | 3099 |
| 1.00 | 3021 | | 3.25 | 3105 |
| 1.25 | 3032 | | 3.50 | 3117 |
| 1.50 | 3043 | | 3.75 | 3122 |
| 1.75 | 3053 | | 4.00 | 3131 |
| 2.00 | 3063 | | 4.50 | 3138 |




**Gambar 4.8** *Kurva pengaturan input terhadap kecepatan motor (rpm)*

Sumbu X : *input* ( V )

Sumbu Y : *output* kecepatan (rpm)

## 5. Penutup

Kesimpulan yang dapat diambil dari penelitian adalah :

1. Perangkat lunak Simulink pada MATLAB mampu mengendalikan kecepatan putaran motor DC.
2. Metode kendali yang diimplementasikan pada sistem pengendalian kecepatan putaran motor DC adalah metode kendali konvensional.
3. Metode penalaan parameter kendali yang diimplementasikan adalah metode *Ziegler-Nichols*.
4. Berdasar hasil pengujian pada bab IV disimpulkan bahwa penggunaan metoda kendali konvensional dapat memperbaiki respon sistem.

**Daftar Pustaka**


[1]Akbar, Agus Taufik.2007.*Kaji Eksperimental Sistem Kendali Kecepatan Motor DC pada Roda Autonomous Robot dengan Metoda Konvensional Berbasis Mikrokontroller ATMEGA8535*. Bandung : Politeknik Manufaktur Bandung.

[2]_____.____.Modul PRAKTIKUM kendali lanjut.www.norture.com.10 Juli 2009.

[3]Barr,Michael. _____. *Introduction to Pulse Width Modulation (PWM)*. NETRINO. 1 Mei 2009.

[4]Prasimax.2009.*Pengendalian Motor DC PWM*.Pengendalian-Motor-DC-PWM.html. 3 Mei 2009.

[5]_____.____.*Pengaturan Kecepatan Motor dengan PC oleh DST-52*(pdf). 3 Mei 2009.

[6]Prasimax.____.*Tutorial1:MengenalMatlab*. PRASIMAX MIKRON 123 Online.3 Mei 2009.

[7]_____. 1999. *Simulink Dynamic System Simulation for MATLAB*(pdf). www.mathworks.com. 2 Maret 2009.

[8]_____.____. *Instrument Control Toolbox 2.8*. www.mathworks.com. 22 Mei 2009.