Sistem Pengendali Robot Bergerak Lurus dengan Kontrol Proporsional Integral Derivatif (PID) Berbasis LabView

Straight-Move Robot Control System with LabView-Based Proportional Integral Derivative (PID) *Control*

Andrean George W, Abdul Rahman
Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Komputer
Universitas Komputer Indonesia Jl. Dipati ukur No 112, Bandung
Email : andreangw7@email.unikom.ac.id

Abstrak - Pengendalian dan monitoring kecepatan putar suatu roda (motor DC) dalam sebuah sistem proses sangat penting perannya dalam implementasi industri. Pengendalian dan monitoring PWM untuk kecepatan putar roda pada sepasang motor DC ini menggunakan perangkat antarmuka komputer dimana dalam industri hal ini diperlukan untuk memudahkan operator dalam mengendalikan dan memonitor kecepatan motor. Agar diperoleh pengontrol yang terbaik, maka dilakukan tuning parameter pengontrol *Proporsional Integral Derifatif* (PID). Dalam tuning ini kita dapat mengetahui nilai dari proporsional gain (Kp), waktu integral (Ti) dan waktu derivatif (Td). Pengontrol PID akan memberikan aksi kepada kontrol motor DC berdasarkan error yang diperoleh, nilai putaran motor DC yang diinginkan disebut dengan set point. Software LabVIEW digunakan sebagai pemonitor, kendali kecepatan motor.

Kata kunci *:* LabView, Motor DC, Arduino, LabView, PID

Abstract *- Control and monitoring of the rotational speed of a wheel (DC motor) in a process system is very important role in the implementation of the industry. PWM control and monitoring for wheel rotational speed on a pair of DC motors uses computer interface devices where in the industry this is needed to facilitate operators in controlling and monitoring motor speed. In order to obtain the best controller, tuning the Integral Derifative (PID) controller parameter is done. In this tuning we can know the value of proportional gain (Kp), integral time (Ti) and derivative time (Td). The PID controller will give action to the DC motor control based on the error obtained, the desired DC motor rotation value is called the set point. LabVIEW software is used as a PE monitor, motor speed control.*

Keyword *: LabView, Motor DC, Arduino, LabView, PID..*

I. PENDAHULUAN

A. Latar Belakang

Sistem kendali pada jaman ini mengalami perkembangan pesat dikarenakan kebutuhan industri yang membutuhkan sistem dengan kecepatan dan akurasi yang tinggi. Seiring berjalannya waktu masyarakat umum juga semakin membutuhkan peralatan elektronik penunjang kehidupan sehari hari yang memiliki kecepatan dan akurasi yang tinggi. Oleh karena itu diperlukan sebuah sistem kendali yang dapat bekerja yang dapat memenuhi tuntutan akan sebuah sistem yang dapat menhadirkan kecepatan dan akurasi yang tinggi. Terdapat beberapa metode sistem pengendali yang ada saat ini. Salah satu metode sistem pegendali yang dapat menghasilkan keluaran yang baik adalah sistem kendali PID. PID adalah sebuah mekanisme kontrol umpan balik yang membutuhkan kendali yang bekerja secara kontinyu [1]. Sistem kendali PID merupakan *tool* standar yang digunakan pada otomasi industri. Fleksibilitas dari kontroler PID memungkinkannya untuk dapat digunakan di banyak aplikasi sistem kendali [2]. Terdapat banyak masalah dalam sistem kendali yang dapat ditangani dengan baik oleh sistem kendali PID. Pengontrol PID menghitung nilai galat *e(t)* sebagai nilai perbandingan dari nilai acuan atau *setpoint* (SP), dan variabel proses (PV), yang akan di koreksi berdasarkan nilai proporsional, integral, dan derivatif (disingkat sebagai PID) sebagaimana nama dari sistem ini. PID ini akan sangat berguna sekali untuk mencapai keluaran yang diinginkan karena dapat mengatasi masalah-masalah perubahan nilai yang fluktuatif dan dengan menciptakan sistem yang responsif terhadap gangguan lainnya. Dengan sistem PID kita dapat

mengatur respon seperti apa yang kita ingin gunakan dalam sistem yang akan di rancang. Sistem kendali PID dapat dengan efektif digunakan untuk sistem linier, tetapi biasanya sulit atau tidak dapat digunakan pada sistem tingkat tinggi dan nonlinier [3].

Sistem kendali PID juga akan digunakan pada pengendalian kecepatan motor DC dalam penelitian ini. Sistem pengendalian kecepatan motor DC merupakan salah satu aplikasi yang paling penting dan paling banyak digunakan pada sistem kendali *non-linear* [4]. Motor DC merupakan peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik. Secara umum, kecepatan putaran poros motor DC akan meningkat seiring dengan meningkatnya tegangan yang diberikan. Dengan demikian, putaran motor DC akan berbalik arah jika polaritas tegangan yang diberikan juga dirubah. [5]. Untuk mengatasi hal ini maka diperlukan suatu perancangan sistem kontrol kecepatan motor DC agar motor DC tersebut bergerak sesuai dengan kecepatan yang diinginkan. Yakni kontroller Proportional Integral Derivatif (PID) yaitu kontrol yang terdiri dari konfigurasi standar Kp, Ki, dan Kd yang nilainya ditentukan / setting agar mendapatkan hasil atau kecepatan yang diinginkan yaitu kecepatan dengan stabilitas yang baik dengan tingkat error dan overshoot (melampaui) yang kecil [6]. Dalam penelitian ini, sistem kontrol PID yang akan dibahas menggunakan prinsip kerja feedback encoder dengan optocoupler berbasis Mikrokontroller ATMega8535 yang akan diaplikasikan pada motor DC tanpa beban. Pembahasan dilakukan dengan mengamati perubahan pada kontrol PID yang disetting pada keypad serta waktu terbaik (Time Sampling) yang diperoleh guna mendapatkan kestabilan dari kecepatan motor DC yang diinginkan [7], sehingga dengan demikian kecepatan putar dari motor dc akan tetap stabil dengan *error* yang minimal pada nilai *set-point* yang telah ditentukan [8].

B. Tinjauan *State of Art*

Sistem kendali PID yang digunakan pada penelitian ini sama dengan sistem kendali PID yang umum digunakan. Namun pada penelitian ini terdapat beberapa software bantuan yang akan digunakan untuk membantu sistem kerja hardware. Pada penelitian ini akan dibuat sebuah aplikasi pada PC menggunakan LABView. Aplikasi ini akan secara realtime berkomunikasi dengan hardware sistem kendali dan berfungsi sebagai *interface* untuk sistem kendali PID [9]. Pada aplikasi inilah user dapat menentukan *set-point* yang diinginkan. Selain itu, seluruh proses kerja dari sistem kendali PID akan ditampilkan pada aplikasi ini, sehingga dengan demikian memudahkan user dalam memantau suhu air (nilai aktual), kestabilan, performa, dan proses kerja dari sistem kendali PID pada penelitian ini [10].
Dalam penelitian ini telah diimplementasikan suatu sistem kontrol motor DC dengan kontroler PID menggunakan mikrokontroler ATmega16. Interface sistem kontrol ini menggunakan Visual Basic 6.0 sehingga memudahkan untuk pengembangan lebih lanjut [11]. Proses identifikasi sistem dilakukan dengan menggunakan sinyal input Pseudo Random Binary Sequence (PRBS) 10 bit dengan variasi sinyal acak sebanyak 1024 data. Dari hasil identifikasi diperoleh fungsi alih model system adalah Selanjutnya dilakukan proses penentuan parameter PID dengan menggunakan metode root locus, yang hasilnya menunjukkan bahwa semua akar berada disebelah kiri bidang s. Sehingga respon yang didapat dari semua pole stabil. Hasil perhitungan parameter PID dengan pole s = -9.4 + j3.2 didapatkan nilai parameter PID terbaik yaitu KP=4.6805, KI=10 dan KD=0.1026 [12]. Dalam penelitian yang sudah pernah dilakukan sebelumnya dapat dilihat beberapa perbedaan. Perbedaan pertama dapat dilihat dari penggunaan software Visual Basic 6.0 dan penggunaan mikrokontroler AT mega 16. Kelebihan penelitian yang kami lakukan dari penelituan sebelumnya adalah dari segi kemudahan. Dengan kita menggunakan software labview itu dapat membuat pengendalian sistem menggunakan PID menjadi lebih praktis dan penggunaan mikrokontroler arduino juga lebih mudah dari mikrokontroler AT mega 16.

C. Tujuan

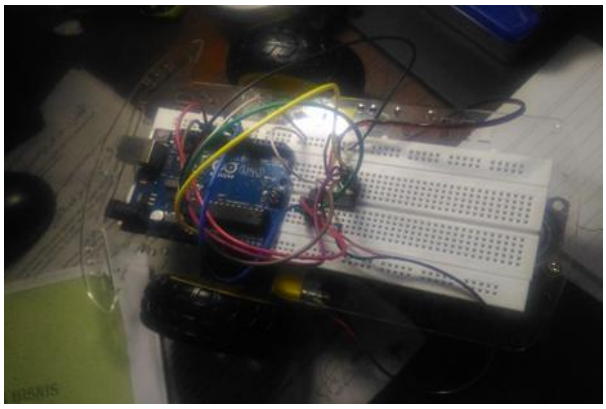
Tujuan dari penelitian ini adalah membuat sebuah sistem pengendalian kecepatan motor dc yang akan mengatur kecepatan dari motor dc sesuai dengan keinginan atau kebutuhan pengguna. Pengguna hanya perlu memasukkan nilai acuan (*set-point*) kecepatan yang diinginkan. Dan setelah itu sistem akan bekerja untuk mencapai nilai *set-point* yang di-*input* oleh *user.* Dan ketika kecepatan yang diinginkan (*set-point)* oleh *user* telah tercapai, maka sistem akan menjaga kestabilan kecepatan, sehingga kecepatan dapat tetap berada pada nilai kecepatan yang diinginkan oleh *user.*

II. METODOLOGI

Penelitian ini merupakan jenis penelitian eksperimen yang dilakukan dengan percobaan pada serangkaian sistem baik *hardware* maupun *software* yang bekerja pada sistem kendali PID. Percobaan juga dilakukan untuk mengetahui seberapa baik *hardware* dan *software* saling terintegrasi satu sama lain pada penelitian ini.

A. Perancangan Alat dan Sistem

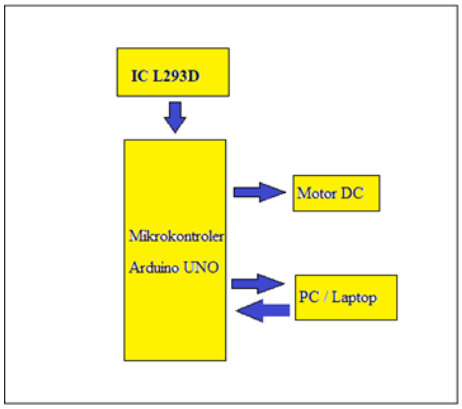
Perancangan sistem kontrol motor DC yaitu sebagai berikut: Kendali 2 buah motor DC pada roda yang akan digunakan adalah dengan metode PWM (Pulse Width Modulation) dengan tegangan input sebesar 12 volt, mikrokontroller bertindak sebagai pengatur kecepatan dengan fitur PWM yang ada pada mikrokontroller pada terhubung ke PIN3, yang merupakan port untuk PWM (PWM0) dengan range nilai antara 255. Mikrokontroller mendapatkan supply 5 volt dari usb connector yang digunakan sebagai line komunikasi serial PC dan mikrokontroller, sedangkan driver motor menggunakan supply 12 volt dari power supply. Bentuk alat (*hardware*) dapat dilihat pada **Gambar 1.**



Gambar 1 Bentuk alat (hardware)

.

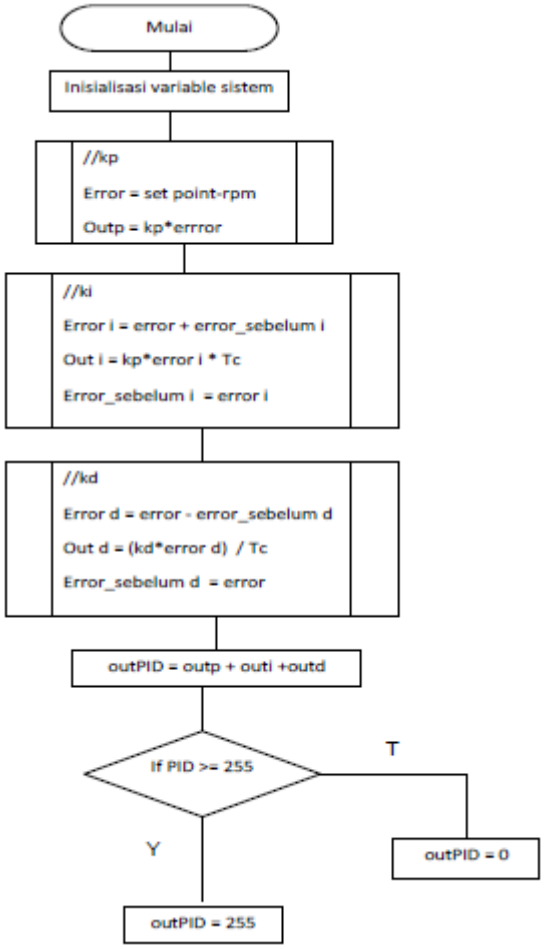
Adapun blok diagram alat kontrol motor DC dapat dilihat pada **Gambar 2.**



Gambar 2. Blok diagram alat kontrol motor DC

B. Pemrograman Kontroler PID

Flowchart program utama kontroler PID meliputi proses inisialisasi, *tuning* parameter, akumulasi error dan perhitungan PID sebagai proses berjalannya motor. Flowchart program utama kontroler PID ditunjukkan dalam **Gambar 3**.



Gambar 3. Flowchart program utama kontroler PID

Pembuatan program kontroler PID ini dilakukan berdasarkan persamaan kontroler PID digital. Kontroler PID digital merupakan bentuk lain dari kontroler PID yang diprogram dan dijalankan menggunakan komputer atau mikrokontroler. Error dan last_erroryang akan digunakan pada perhitungan aksi kontroler PID. Setiap satu kali looping program, error akan diperbaharui dengan data yang diambil dari sensor, dan sebelumnya akan disimpan di last_error. Keluaran dari perhitungan program kontroler PID ini adalah nilai PWM. *Setpoint* (SP) adalah suatu prameter nilai acuan atau nilai yang diinginkan. *Present Value* (PV) adalah nilai pembacaan sensor saat itu atau variabel terukur yang di umpan balikan oleh sensor.

C. Tuning Parameter

Tuningparameter berfungsi untuk melakukan pengesetan terhadap parameter-parameter PID (Kp, Ki dan Kd) dan juga parameter-parameter lainya sepertitime sampling, penentuan set point, dll. Langkah metode tersebut ialah sebagai berikut:

a) Buat suatu sistem loop tertutup dengan kontroler P dan plant di dalamnya,
b) Kemudian hanya dengan menggunakan tindakan kontrol proporsional,dengan Ki=0, Kd=0harga ditingkatkan dari nol ke suatu nilai kritis Kcr, disini mula-mula keluaran memiliki osilasi yang berkesinambungan.
c) Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis Kcr dan periode Pcr dapat ditentukan.
d) Menghitung nilai Kp, Ti dan Td sesuai dengan aturan dari Ziegler-NicholsyaituKp=0.6$xKcr$,Ti=0.5$xPcr$dan Td=0.125$xPcr$
e) Nilai Ki dan Kd didapatkan dengan menggunakan perhitungan sebagai berikutKi=$KpTi$danKd=$KpxTd$

D. Ziegler-Nichols Metode Tuning:

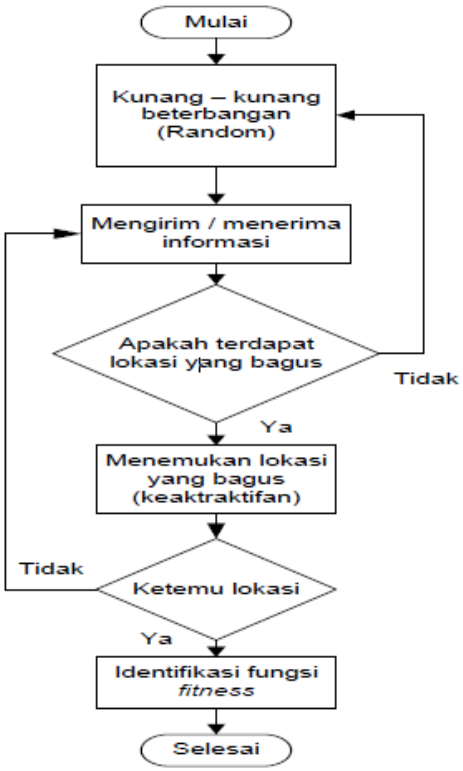
Ada dua metode untuk penentuan parameter kontroler PID Ziegler-Nichols disebut tala aturan. Tetapi metode yang diterima secara luas untuk tuning kontroler PID adalah metode sederhana. Pertama, mengatur controller ke mode P saja. Berikutnya, mengatur gain dari controller (Kp) ke nilai yang kecil. Akhirnya, menyesuaikan sampai respon diperoleh yang menghasilkan osilasi terus menerus. Hal ini dikenal sebagai gain tertinggi (Kp) atau ditunjukkan bahwa periode osilasi dikenal sebagai periode utama (Ku). Langkah-langkah yang diperlukan untuk metode yang diberikan di bawah: Koefisien integral (Ki) dan derivatif (Kd) harus mengatur ke nol.Secara bertahap meningkatkan koefisien proporsional dari nol sampai sampai sistem hanya mulai berosilasi terus menerus. Koefisien proporsional pada titik ini disebut gain tertinggi.

D. Firefly Algorithm

Firefly Algorithm (FA) adalah sebuah algoritma metaheuristik yang terinspirasi dari perilaku berkedip kunang-kunang. Algoritma ini dikembangkan oleh Dr Xin-She Yang di Universitas Cambridge pada tahun 2007. Ada banyak spesies kunang – kunang dan sebagian besar menghasilkan cahaya dalam durasi yang pendek dan memiliki ritme tertentu. Cahaya dihasilkan dari proses *bioluminiscene.* Terdapat dua fungsi fundamental dari cahaya tersebut, yaitu untuk menarik perhatian kunang – kunang yang lain (komunikasi) dan untuk bertahan dari serangan *predator.* Fungsi lainnya untuk mekanisme peringatan bahaya. Cahaya ini dimiliki oleh *firefly,* baik jantan maupun betina. (Yang, X, S (2009)

Dr Xin-She Yang mengembangkan *Firefly Algorithm* berdasarkan kebiasan dan pola kehidupan *firefly* tersebut. Dalam merumuskan *Firefly Algorithm*, Dr Xin-She Yang mengasumsikan beberapa aturan :

1. Semua *firefly* itu berjenis kelamin satu sehingga kunang– kunang akan tertarik pada *firefly* lain terlepas dari jenis kelamin mereka.
2. Daya tarik sebanding dengan tingkat kecerahan *firefly*, maka *firefly* dengan tingkat kecerahan lebih rendah akan tertarik dan bergerak ke *firefly* dengan tingkat kecerahan lebih tinggi, kecerahan dapat berkurang seiring dengan bertambahnya jarak.
3. Jika tidak ada *firefly* yang lebih terang dari *firefly* yang diberikan, maka kunang-kunang ini akan bergerak secara random. *Flowchart* dari *Firefly algorithm* dapat dilihat pada **Gambar 4**.



Gambar 4. Diagram alir *Firefly Algorithm* (FA)

Parameter PID yang ditala oleh FA adalah Kp, Ki dan Kd. Adapun untuk diagram alir proses penalaan parameter PID dengan menggunakan metode Firefly Algorithm (CSA) ditunjukkan oleh flowchart pada Gambar 4. Gambar 5 menunjukkan pemodelan motor DC pada Simulink Matlab 2015, tanpa kontrol, dengan Bee Colony, Cuckoo Search dan PID Firefly.

Untuk menjalankan algoritma firefly dibutuhkan beberapa parameter, yang disebutkan pada Tabel 2. Algoritma firefly dibuat menggunakan software Matlab (m.files) dan pemodelan motor menggunakan labview. Adapun data parameter-parameter firefly adalah berikut,

Parameter Nilai :

a. Alpha : 0.25
b. Beta : 0.2
c. Gamma : 1
d. Dimensi : 80
e. Jumlah Firefly : 80
f. Iterasi Maksimum : 50

E. Perancangan Software

Pada penelitian ini, akan digunakan algoritma/program PID berbasis Labview. Dimana akan terdapat *Graphic User Interface* yang akan digunakan *user*/pengguna untuk memasukkan nilai suhu yang dinginkan (*set point*). Nilai *set point* ini nantinya akan dibandingkan dengan nilai suhu yang terbaca oleh sensor (nilai aktual) untuk mengetahui nilai *error* antara *set-point* dan nilai aktual. Sistem kendali PID menghitung nilai galat *e(t)* sebagai nilai perbandingan dari nilai acuan atau *Setpoint*(SP) dan variabel proses(PV). Sistem PID ini dapat di aplikasikan kedalam aplikasi *Labview*.

III. HASIL DAN PEMBAHASAN

A. Pengujian output Sensor Suhu DS18B20

Prosedur pengujian dilakukan dengan menghubungkan rotary encoderdengan motor DC kemudian memasukkan program ke dalam MK sesuai RPM yang ingin kita ukur. Motor DC dinyalakan dan putaran rotary encoder diukur menggunakan tachometer.Data input yang dimasukkan ke dalam mikrokontroletor untuk menggerakkan adalah sebesar 200 rpm sehingga rotary encoder akan berputar sesuai dengan putaran motor DC yaitu 200 rpm. Sedangkan tachometer yang digunakan untuk mengukur mengukur rotary encoder mencacat hasil 190 rpm.

B.Pengujian Motor DC

Tujuan dari pengujian ini adalah untuk mengetahui output dari driver motor apabila diberi input yang berbeda-beda.

Tabel 1.Hasil Pengujian Kecepatan PWM

PWM	V MK (Volt)	V Motor (Volt)	Rpm
10	0.215	0.28	0.2
20	0.41	0.61	1.8
30	0.41	0.61	11.3
40	0.795	1.29	25.2
50	0.98	1.63	35
60	1.17	2.08	41
70	1.35	2.79	46
80	1.54	2.87	48.8
90	1.75	3.45	52
100	1.35	4.6	55.6
110	1.54	5.5	58.8
120	1.75	6.2	60.8
130	1.9	6.7	64.2
140	2.13	7.4	67
150	2.32	7.77	72.2
160	3.29	8.2	74.6
170	3.48	8.54	79.2
180	3.66	9.2	81.6
190	3.87	9.53	88
200	4.05	9.9	90.8
210	4.25	9.95	100.4
220	4.42	10.12	116.4
230	4.6	10.32	124
240	4.78	10.54	128
250	4.82	10.6	128.5
255	4.9	10.67	129.2

C. Pengujian Keseluruhan

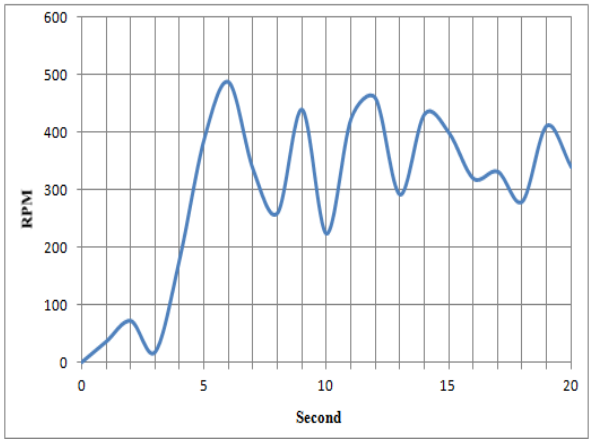
1) Pengujiaan Tuning Kontroler PID

Tahap-Tahap Pengujian PID:

1. Menjalankan sistem dan alat
2. Mengamati tampilan grafik dari sensor *rotary encoder*.
3. Apabila grafik belum berbentuk osilasi kesinambungan, naikkan nilai Kp, harga ditingkatkan dari nol ke suatu nilai kritis *K*cr.

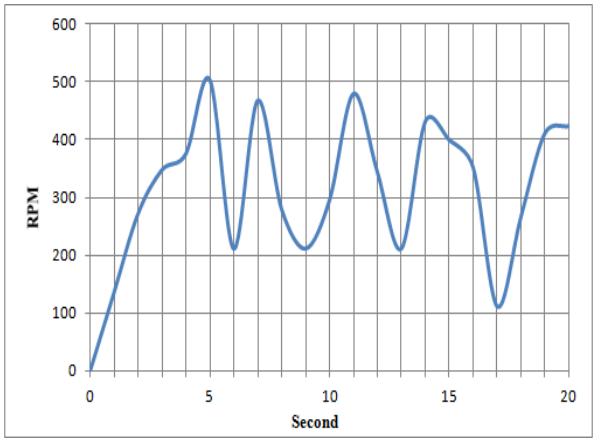
4. Apabila grafik sensor *rotary encoder* sudah membentuk osilasi kesinambungan, hitung nilai Kcr dan Pcr.
5. Hitung nilai Ti dan Td dengan menggunakan nilai Kcr dan Pcr.
6. Hitung nilai Ki dan Kd dengan menggunakan nilai Ti dan Td.
7. Dengan menggunakan nilai Kp, Ki dan Kd yang telah didapat, amati kecepatan putaran.

Pengujian terhadap kontrol kecepatan dengan metode osilasi Ziegler-Nichols dimulai dengan memberikan nilai 0 pada parameter Ti dan Td. Sedangkan nilai Kp dinaikkan sedikit demi sedikit hingga didapatkan grafikyang berosilasi berkesinambungan. Hasil pengujian untuk respon kecepatan motordengan menggunakan kontroler proporsional dengan nilai 1(Kp=1) dapat dilihat pada **Gambar 4**.



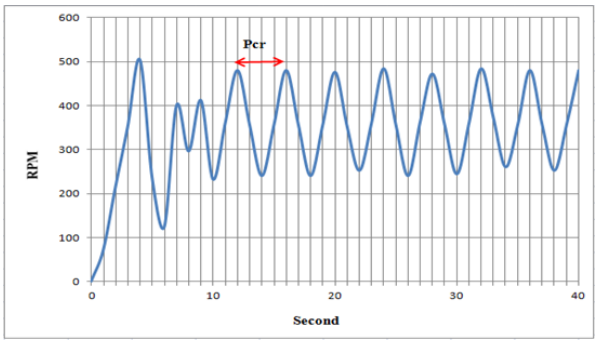
Gambar 4. Grafik Respon kecepatan motor dengan Kp= 1

Terlihat pada grafik berosilasi tetapi belum berkesinambungan sehingga masih perlu ditambahkan nilai Kp. Hasil pengujian dengan menggunakan dengan nilai 2 (Kp=2) dapat dilihat pada **Gambar 5**.



Gambar 5. Grafik Respon kecepatan motor dengan Kp= 2

Pengujian dengan menggunakan nilai Kp=2 terlihat bahwa respon belum membentuk osilasi berkesinambungan. Hasil pengujian untuk respon kecepatan motor dengan menggunakan kontroler proporsional dengan nilai 3 (Kp=3) dapat dilihat **pada Gambar 6.**



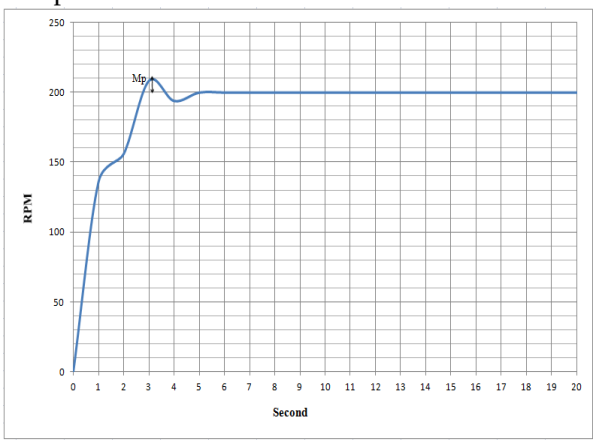
Gambar 6. Grafik Respon kecepatan motor saat terjadi osilasi kesinambungan dengan Kp= 3

Pada pengujian dengan menggunakan nilai Kp=3 terlihat bahwa respon sudah mengalami osilasi kesinambungan pada detik ke 10. Terlihat bahwa pada saat kontroler proporsional bernilai 3 dapat membentuk osilasi berkesinambungan dibandingkan dengan kp=2 Sehingga dari grafik diatas dapat dihitung nilai Kcr dan Pcr yaitu,

- Kcr=3
- Pcr=(16−12) x $Time\ Sampling$=4 x 1 s=4 s
- Pcr=(16−12) x $Time\ Sampling$=4 x 1 s=4 s
- Kp=0,6 x Kcr=0,6 x 3=1,8
- Ti=0,5 x Pcr=0,5 x 4 s=2 s
- Td=0,125 x Pcr=0,125 x 4 s=0,5 s
- Ki=$KpTi$=1,82=0,9
- Kd=$Kpx\ Td$=1,8 x 0,5=0,9

2) Set point 200 rpm

1. Tanpa beban



Gambar 7. Grafik Respon kecepatan motor *screw conveyor* pada set point 200 rpm tanpa beban

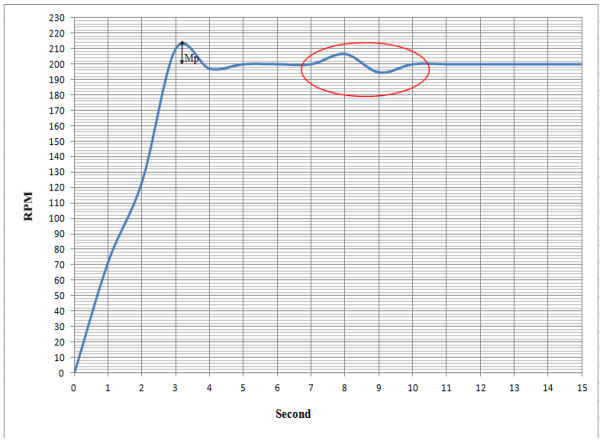
Dari Gambar 7 didapatkan nilai time settling atau ts dan error steady state, berikut akan dijelaskan pengertian time settling atau ts dan error steady state, beserta perhitungannya.

a. Time settling pada kecepatan 200 rpm tanpa beban didapatkan Ts= 4s

b. Error steady state yang didapatkan dari pengujian dengan set adalah: ess= (203,2−200)200x 100 %=1,6 %

c. Maximum Overshoot (Mp) merupakan Nilai tertinggi dari grafik adalah 208,4 maka: Mp= (208,4−200)200 x 100% = 4,2 %

2. Beban 500 g



Gambar 8. Grafik Respon kecepatan motor *screw conveyor* pada set point 200 rpm 500 g

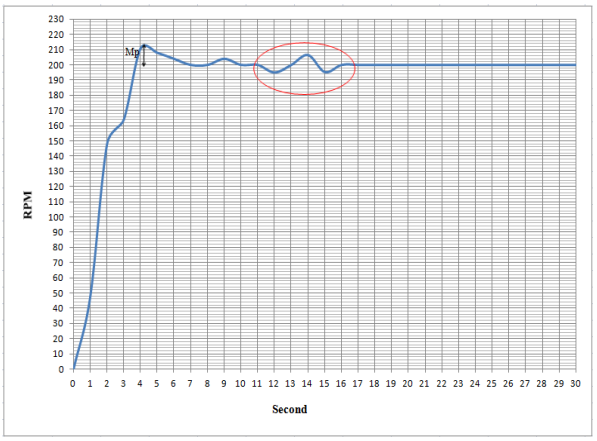
Dari Gambar 8 didapatkan nilai time settling atau ts dan *error steady state* (lingkaran merah) yang didapat karena adanya perubahan putaran motor yang diakibatkan oleh adanya adonan roti yang masuk ke dalam celah pitch screw yang berbeda, berikut akan dijelaskan pengertian time settling atau ts dan *error steady state*, beserta perhitungannya.

a. Time settling pada kecepatan 200 rpm dengan beban 1 kg didapatkan Ts= 4s

b. Error steady state yang didapatkan dari pengujian dengan set adalah: ess= (206,2−200)200x 100 %=3,1 %

c. Maximum Overshoot (Mp) merupakan Nilai tertinggi dari grafik adalah 210,3 maka: Mp= (210,3−200)200 x 100% = 5,15 %

3. Beban 1 kg



Gambar 9. Grafik Respon kecepatan motor *screw conveyor* pada set point 200 rpm 1kg

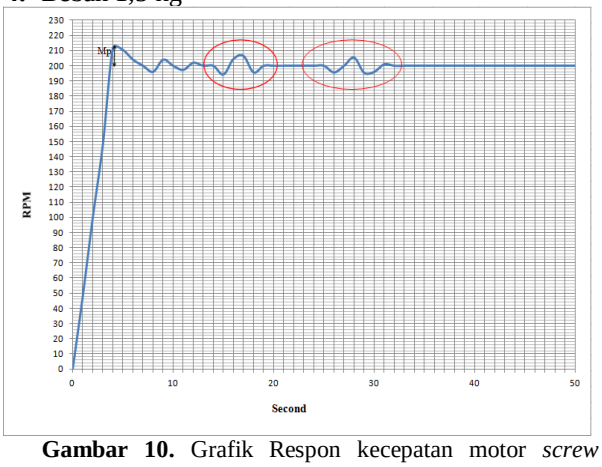
Dari **Gambar 9** didapatkan nilai time settling atau ts dan *error steady state* (lingkaran merah) yang didapat karena adanya perubahan putaran motor yang diakibatkan oleh adanya adonan roti yang masuk ke dalam celah pitch screw yang berbeda, berikut akan dijelaskan pengertian time settling atau ts dan *error steady state*, beserta perhitungannya.

a. Time settling pada kecepatan 200 rpm dengan beban 2 kg didapatkan Ts= 5s

b. Error steady state yang didapatkan dari pengujian dengan set adalah: ess= (206,7−200)200x 100 %=3,35 %

c. Maximum Overshoot (Mp) merupakan Nilai tertinggi dari grafik adalah 210,9 maka: Mp= (210,9−200)200 x 100% = 5,45 %

4. Beban 1,5 kg



Gambar 10. Grafik Respon kecepatan motor *screw conveyor* pada set point 200 rpm 3kg

Dari **Gambar 10** didapatkan nilai time settling atau ts dan *error steady state* (lingkaran merah) yang didapat karena adanya perubahan putaran

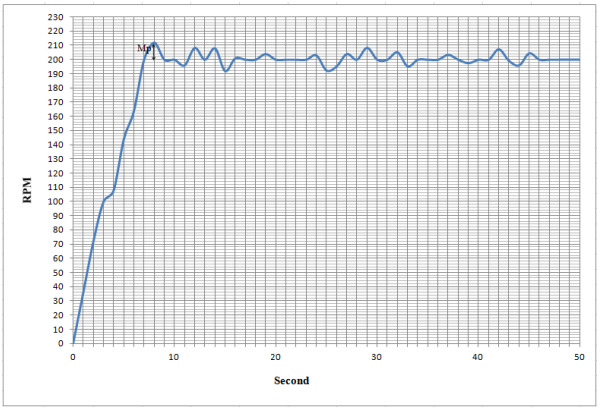
motor yang diakibatkan oleh adanya adonan roti yang masuk ke dalam celah pitch screw yang berbeda, berikut akan dijelaskan pengertian *time settling* atau ts dan *error steady state*, beserta perhitungannya.

a. Time settling pada kecepatan 200 rpm dengan beban 3 kg didapatkan Ts= 5s

b. Error steady state yang didapatkan dari pengujian dengan set adalah: ess= (207,3−200)200x 100 %=3,65 %

c. *Maximum Overshoot* (Mp) merupakan Nilai tertinggi dari grafik adalah 211,6 maka: Mp= (211,6−200)200 x 100% = 5,80 %

5. Beban 2 kg



Gambar 11. Grafik Respon kecepatan motor *screw conveyor* pada set point 200 rpm 2kg

Dari Gambar 11 didapatkan nilai *time settling* atau ts dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau ts dan *error steady state*, beserta perhitungannya.

a. Time settling pada kecepatan 200 rpm dengan beban 4 kg didapatkan Ts= 7s

b. Error steady state yang didapatkan dari pengujian dengan set adalah: ess= (208,2−200)200x 100 %=4,1 %

c. *Maximum Overshoot* (Mp) merupakan Nilai tertinggi dari grafik adalah 212, maka: Mp= (212,3−200)200 x 100% = 6,15 %

Jika dilihat dari Gambar 11 terdapat banyak *error steady state* yang mengakibatkan putaran motor susah mencapai set point, hal ini dikarenakan beban yang dapat diatur PID adalah terbatas sehingga pada percobaan set point 200 dengan beban berikutnya disarankan diatur kembali kontroler PID dengan cara mengubah parameter Kp, Ki dan Kd.

Pengujian ini ditentukan memiliki prosentase kesalahan Ess yang diperbolehkan adalah 5% berdasarkan sistem yang dinyatakan. Dengan demikian nilai toleransi yang diperbolehkan pada pada 200 rpm adalah 10 rpm. Berikut merupakan data hasil pengujian performansi sistem pada **Tabel 2**.

Tabel 2. Tabel Data Performansi Sistem set point 200 rpm

Beban (g/kg)	Motor (rpm)	Settling time (ts)	Overshoot (%)	Ess (%)
0	200	4 detik	4,2	1,6
1	200	4 detik	5,15	3,1
2	200	5 detik	5,45	3,35
3	200	5 detik	5,80	3,65
4	200	7 detik	6,15	4,1

Setelah memasukkan beberapa parameter tersebut di Tabel 2, maka selanjutnya algoritma Firefly bisa dijalankan untuk optimasi nilai PID dari controller. Nilai yang tepat akan sangat mempengaruhi kinerja respon motor DC yang didesain pada penelitian ini. Algoritma Firefly membutuhkan proses perhitungan sampai menemukan nilai yang optimal. Gambar 6 menunjukkan grafik konvergensi optimasi nilai PID menggunakan algoritma firefly. Konvergensi adalah suatu nilai fitness function yang menjabarkan kriteria optimal dari suatu masalah optimasi.

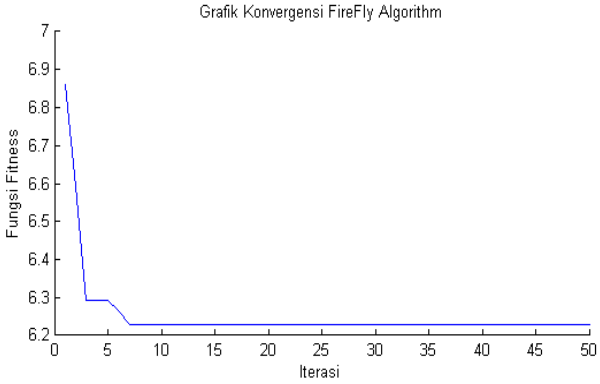
Pada Gambar 12 menunjukkan grafik konvergensi optimasi nilai PID menggunakan firefly, di mana berdasarkan grafik terlihat algoritma firefly tidak membutuhkan waktu yang lama dalam melakukan proses optimasi, hal tersebut terlihat pada iterasi ke 7 algoritma sudah menemukan nilai PID yang optimal dengan nilai fitness sebesar 6.2269. Untuk hasil selengkapnya dapat dilihat pada **Tabel 3**.

Tabel 3. Hasil Optimasi dengan Firefly

Total number of iterations=50
fmin = 6.2269
nbest = 40000 3.1641 1.0000
kp_ff = 40
ki_ff = 3.1641
kd_ff = 1

Hasil optimasi Firefly didapatkan nilai *fitness function* sebesar *6.2269*, dengan 50 kali iterasi, nilai nbest merupakan firefly terbaik, yang di

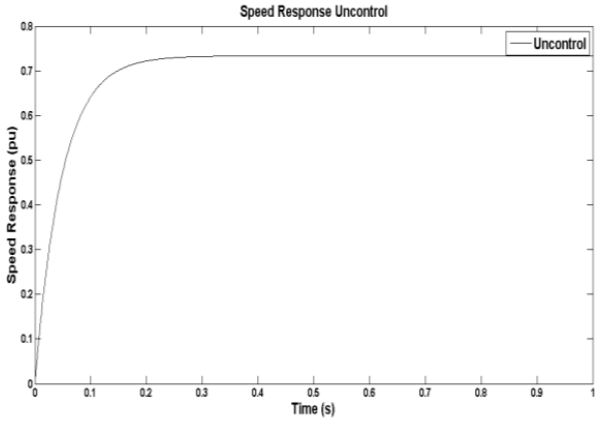
mana diketahui sebagai hasil optimasi parameter PID, yaitu Kp, Ki dan Kd. Tabel 3 menunjukkan nilai hasil optimasi parameter PID ditala oleh *Firefly*. Sebagai pembanding digunakan algoritma Bee Colony. Bee Colony merupakan salah satu algoritma cerdas yang dalam hal ini digunakan dari penelitian sebelumnya. Bee Colony merupakan algoritma yang bekerja berdasarkan perilaku lebah dalam mencari sumber makanan. Sumber makanan di sini diasumsikan sebagai nilai PID yang akan dioptimasi.



Gambar 12. Grafik Konvergensi Optimasi Kontrol PID Motor DC dengan *Firefly Algorithm*

6. Respon Kecepatan Motor DC tanpa Controller

Simulasi pertama adalah simulasi open loop Motor DC tanpa *controller*. Berikut hasil simulasi.

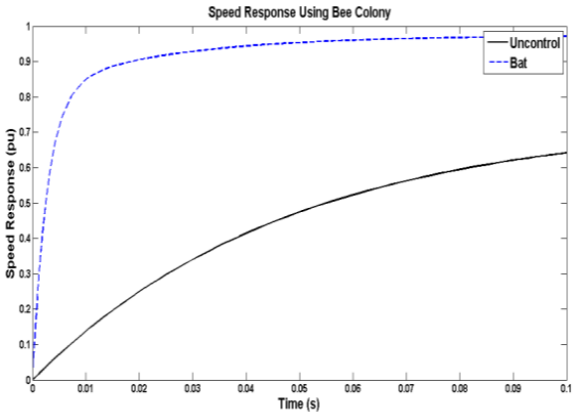


Gambar 13. Respon Kecepatan Motor DC tanpa kontrol, untuk t=1s.

Gambar 13 menunjukkan hasil simulasi tanpa kontroler, didapatkan respon kecepatan motor DC yang sangat tinggi, hal ini dikarenakan sistem tidak ada umpan balik, sehingga motor bekerja tanpa ada batasan dan untuk sistem yang seperti ini sangat dihindari. Untuk itu sangat diperlukan desain sistem kontrol yang tepat dengan penambahan kotroler PID pada motor DC, sehingga kecepatan yang dihasilkan dapat dikontrol sesuai dengan beban yang dikopel oleh motor DC.

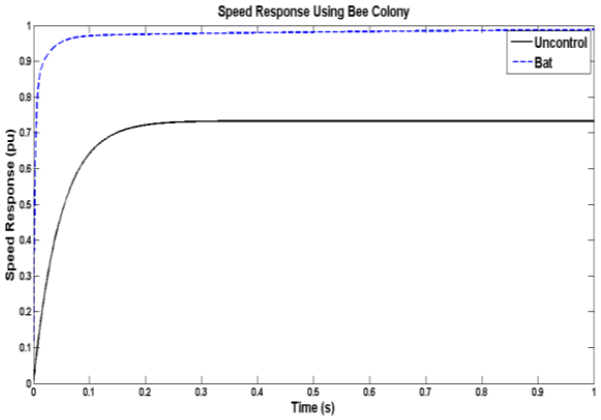
7. Respon Kecepatan Motor DC dengan PID Bee Colony

Simulasi yang kedua adalah kontrol motor DC dengan PID Bee Colony, berikut hasil simulasinya pada **Gambar 14** dan **15**.



Gambar 14. Respon Kecepatan Motor DC dengan PID Bee Colony, t=0.1s

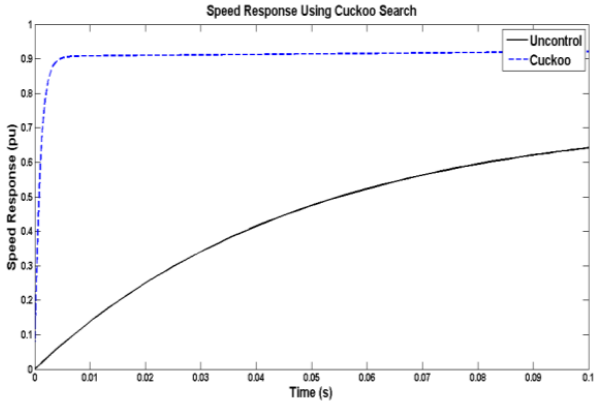
Untuk lebih jelasnya berikut untuk t=1s.



Gambar 15. Respon Kecepatan Motor DC dengan PID Bee Colony, t=1s

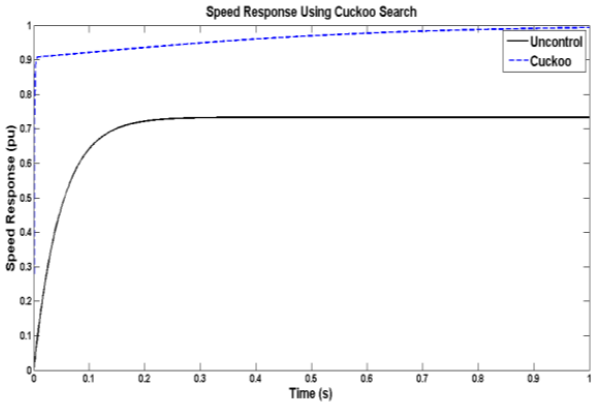
Gambar 14 dan **15** menunjukkan grafik respon frekuensi motor DC dengan PID Bee Colony. Dari grafik di atas didapatkan *settling time* yang semakin baik dibanding dengan tanpa control, namun kinerja PID pada system ini masih bisa dioptimalkan dengan penalaan yang tepat. Dari grafik dapat dilihat, respon system sudah sangat baik dalam mencapai kondisi steady state sebelum t=1s.

8. Respon Kecepatan Motor DC dengan PID Cuckoo Search Algorithm Simulasi yang ketiga adalah kontrol motor DC dengan menggunakan PID Cuckoo, berikut hasil simulasinya.



Gambar 16. Respon Kecepatan Motor DC dengan PID Cuckoo, t=0.1s

Untuk lebih jelasnya berikut untuk t=1s.



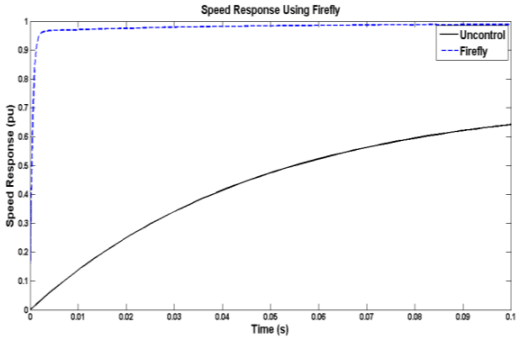
Gambar 17. Respon Kecepatan Motor DC dengan PID Cuckoo t=1s

Gambar 16 dan 17 menunjukkan grafik respon frekuensi motor DC dengan PID Cuckoo. Dari grafik di atas didapatkan *settling time* yang sangat cepat dibanding dengan metode bee colony, di mana sistem sudah berada pada kondisi *steady* sebelum t=1s. Kinerja PID pada system ini masih bisa untuk dioptimalkan.

9. Respon Kecepatan Motor DC dengan PID Firefly

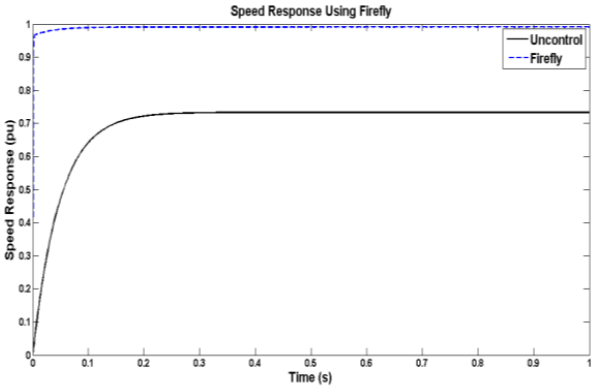
Simulasi yang keempat menggunakan metode yang diusulkan pada penelitian ini, yaitu menggunakan metode firefly. . Dari grafik 18 dan 19 didapatkan *settling time* yang sangat cepat dengan menggunakan firefly, dibanding dengan metode bee colony dan cuckoo, di mana sistem lebih cepat berada pada kondisi *steady.*

Berikut hasil simulasi percobaan motor dc pada **Gambar 18** dan **Gambar 19**.



Gambar 12. Respon Kecepatan Motor DC dengan PID Firefly, t=0.1s

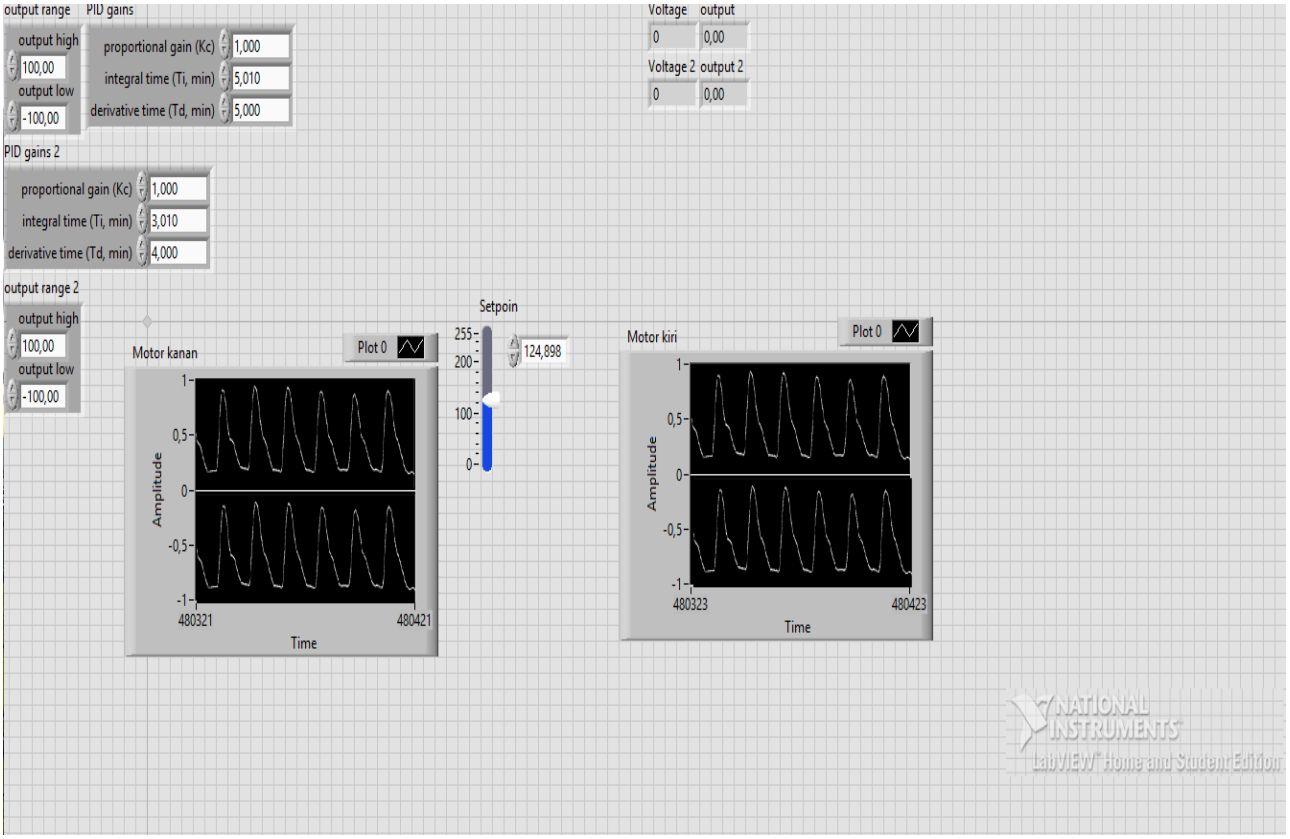
Untuk lebih jelasnya berikut untuk t=1s.



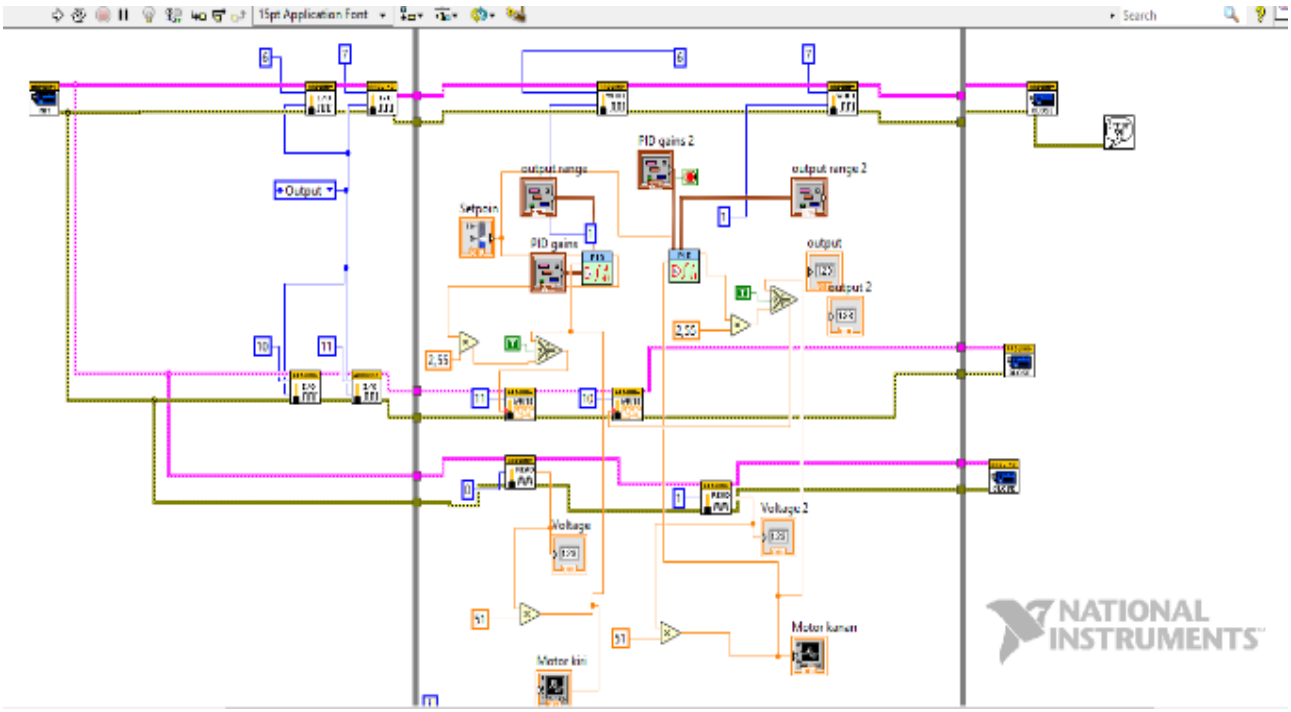
Gambar 13. Respon Kecepatan Motor DC dengan PID Firefly, t=1s

Gambar 12 dan **13** menunjukkan grafik respon frekuensi motor DC dengan PID Firefly. Dari grafik 12 dan 13 didapatkan *settling time* yang sangat cepat dengan menggunakan firefly,.

Sehingga didapatkan nilai parameter Kp= 1,8, Ki= 0,9 dan Kd= 0,9 Pengaturan speed / kecepatan motor adalah dengan menerapkan metode pengaturan kecepatan menggunakan PWM, yaitu metode dengan merubah nilai duty cycle pada driver motor, dengan begitu tegangan yang dialirkan pada motor dapat diatur , PWM yang digunakan pada penelitian ini menggunakan fitur timer 0 pada mikrokontroller dengan resolusi 8 bit (0 s/d 255). Disini juga kita menguji dengan menggunakan program delay pada labview dimana pada waktu setiap 5 detik sekali roda akan berhenti berputar lalu roda akan berbelok 90 derajat ke arah kiri begitu seterusnya hingga robot kembali pada posisi semula dan membentuk sebuah persegi. Tampilan gambar program dan blok diagram pada software labview dapat dilihat pada **Gambar 20** dan **Gambar 21.**



Gambar 20. Tampilan gambat program pada software labview



Gambar 21. Tampilan gambar blok diagram pada software labview

IV. PENUTUP

A. Kesimpulan

Kesimpulan yang didapat dari hasil pengujian adalah sebagai berikut.

1. Sistem pada penelitian ini dapat bekerja dengan baik memenuhi tujuan dari penelitian ini.
2. Hasil pengujian terhadap aplikasi kontrol PID menunjukkan bahwa PID Ziegler Nichols metode osilasi dapat digunakan sebagai kontrol kecepatan motor DC.
3. Penggunaan kontroler PID dengan metode osilasi Ziegler-Nichols memberikan nilai parameter yang sesuai dengan sistem dan memberikan respon yang lebih cepat untuk mencapai keadaan steady state. Dengan parameter PID yang dipakai adalah (Kp=1 Ki=0,9 Kd=0,9).

B. Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut di sarankan:Pembuatan mekanik yang baik dan lebih presisi, akan membuat alat semakin stabil dalam pergerakan dan kecepatanya.

DAFTAR PUSTAKA

[1] Akyuz I H, Yolacan E, Ertunc H M and Bingul Z 2011 PID and state feedback control of a single-link flexible joint robot manipulator. *International Conference on Mechatronics*, pp. 409-414

[2] Chang W D 2009 PID control for chaotic synchronization using particle swarm optimization. *Chaos, Solutions and Fractals*, **39** pp. 910-917

[3] Ptil A B and Salunkhne A V 2008 Adaptive Neuro Fuzzy Controller for Process Control System. *IEEE Region 10 and the Third International Conference on Industrial and Information Systems*

[4] Jha N, Singh U, Saxena T K and Kapoor A 2011 Online Adaptive Control for Non Linear Processes Under Influence of External Disturbance. *IJAE*, **2**(2) pp. 36-46

[5] Mote T P and Lokhande S D 2012 Temperature Control System Using ANFIS. *International Journal of Soft Computing and Engineering*, **2**(1) pp. 156-161

[6] Jung H M, Kim J H, Lee B K, and Yoo D W 2010 A New PWM Dimmer Using Two Active Swtiches fot AC LED Lamp. *The 2010 International Power Electronics Conference*, pp. 1547-1551

[7] Pengfei L, Jiakun L and Junfeng J 2010 Wireless temperature monitoring system based on the ZigBee Technology. *2010 2nd International Conference on Computer Engineering and Technology*, **1** pp. 160-163

[8] ARIA, MUHAMMAD. PID control of a three-degrees-of-freedom model helicopter. *Majalah Ilmiah UNIKOM*, 2011, 9.2: 207-214.

[9] Yang J H and Bi X Y 2010 High-precision Temperature Control System Based on PID Algorithm. *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, **12** pp. 568-571

[10] Li H L and Hu A P 2012 A Direct AC-AC Converter for Inductive Power-Transfer Systems. IEEE TRANSACTIONS ON POWER ELECTRONICS, 27 pp. 661-668

[11] Enjeti, P. N., & Choi, S. (1993, March). An approach to realize higher power PWM AC controller. In Proceedings Eighth Annual Applied Power Electronics Conference and Exposition, (pp. 323-327). IEEE.

[12] Vainio, O., & Ovaska, S. J. (1996). Digital filtering for robust 50/60 Hz zero-crossing detectors. IEEE Transactions on Instrumentation and Measurement, 45(2), 426-430.