

Rancang Bangun Dongkrak Elektrik Berbasis Pengendali *Proportional Integral Derivative (PID)*

A. Asni B¹, Mayda Waruni²

Abstract— *Proportional Integrator Derivative (PID) controller, which aims to set a variable value with stable and precision in an instrument system using a feedback (closed loop) on the system. RPM readings rotation of DC motor that drives the screw jack to be used as a reference control. After implementing the motor rotation speed control system with PID controller jack using the 3-point setting of 60, 50 and 40 RPM. The measurement results show the average speed of each point setting. When the set point average of 60 RPM rotation speed of the motor jack to 56 RPM with an average of 4 RPM below the set point. When the set point average of 50 RPM rotation speed of the motor jack to 47 RPM with an average of 3 RPM below the set point. And when the set point average of 40 RPM rotation speed of jack motor to 37 RPM with an average of 3 RPM below the set point*

Intisari— *Pengendali Proporsional Integrator Derivative (PID) bertujuan untuk mengatur suatu nilai variabel dengan stabil dan presisi pada suatu sistem instrumen dengan menggunakan suatu umpan balik (closed loop) pada sistem tersebut. Pembacaan RPM putaran motor DC yang menggerakkan ulir dongkrak akan digunakan sebagai referensi kontrol.*

Setelah mengimplementasikan sistem kontrol kecepatan putaran motor dongkrak dengan pengendali PID menggunakan 3 setting point yaitu 60, 50 dan 40 RPM didapatkan suatu hasil kecepatan rata-rata tiap setting point. Ketika setting point 60 RPM rata-rata kecepatan putaran motor dongkrak menjadi 56 RPM dengan rata-rata 4 RPM dibawah setting point. Kemudian setting point 50 RPM rata-rata kecepatan putaran motor dongkrak menjadi 47 RPM dengan rata-rata 3 RPM dibawah setting point. Terakhir setting point 40 RPM rata-rata kecepatan putaran motor dongkrak menjadi 37 RPM dengan rata-rata 3 RPM dibawah setting point.

Kata Kunci— *Dongkrak Elektrik, PID, Proportional Integrator Derivative*

I. PENDAHULUAN

Adanya kendaraan merupakan sebuah anugerah yang tak ternilai bagi kita semua sebagai pemakainya salah satunya yaitu memudahkan bagi kita semua untuk bepergian dari satu tempat ke tempat lain hanya dalam tempo yang relatif singkat. Kendaraan seperti mobil dan motor tidak selamanya selalu berfungsi dengan baik, pasti suatu saat dapat mengalami kerusakan. Kerusakan itu terjadi bisa karena kurangnya perawatan dan juga karena musibah seperti terjadi kecelakaan atau bocornya ban kendaraan. Pada mobil dan truk yang

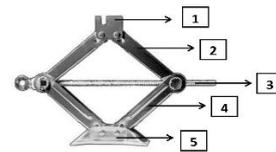
memiliki kapasitas besar, diperlukan alat khusus untuk mengangkat beban mobil tersebut. Alat yang sering digunakan adalah berupa dongkrak. Ada pun dongkrak bermacam – macam diantaranya ada yang berupa hidrolik dan ulir, namun pada umumnya mobil kelas 1400 cc menggunakan dongkrak ulir. Prinsip penggunaan dongkrak ulir menggunakan ulir dengan prinsip kerjanya dengan memutar *knop* agar ulir berputar sehingga penampang terangkat sampai mengangkat beban, namun saat ini masih menggunakan pemutar manual dan membutuhkan tenaga yang cukup besar[1]

Pada penelitian ini akan dibuat suatu model dongkrak elektrik sederhana dengan menggunakan pengendali yang berbasis pengendali PID. Dengan menggunakan pengendali PID, kecepatan motor akan terjaga stabil ketika mengangkat beban berupa mobil atau truk. Selain itu, parameter lain yang perlu diperhatikan adalah posisi ketinggian. Posisi ketinggian akan dijadikan sebagai *input*-an oleh pengguna dengan keypad, parameter tersebut kemudian dikirim ke mikrokontroler dan diolah, hasil pengolahan dari mikrokontroler tersebut dikirim ke *Driver* motor DC. Dongkrak akan bekerja pada referensi tinggi yang telah ditentukan oleh pengguna.

II. DASAR TEORI

2.1 Dongkrak Ulir

Dongkrak ulir merupakan dongkrak yang ringan dan sederhana digunakan. Ukurannya beragam. Ukuran kecil digunakan pada beberapa mobil sedan atau *city car*. Dongkrak ini dapat dioperasikan secara mekanik dan mempunyai beberapa manfaat dalam penggunaannya[1].



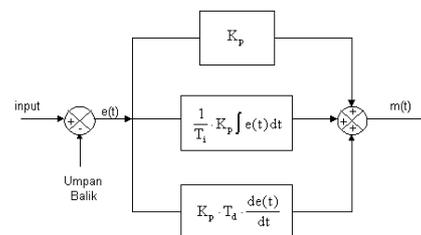
Gambar 2. 2 Bagian Dongkrak Ulir

Bagian dari dongkrak ulir adalah sebagai berikut.

1. Penyangga atas
2. Lengan atas
3. Poros ulir
4. Lengan bawah
5. Kaki Penyangga

2.2 Pengendali PID[2]

Setiap kekurangan dan kelebihan dari masing-masing pengendali P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi pengendali proporsional, integral dan *derivative* (pengendali PID).



Elemen-elemen pengendali P, I dan D masing-masing

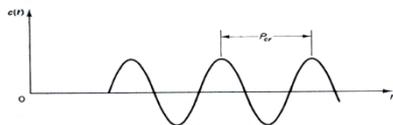
Gambar 2. 1 Diagram Blok Pengendali PID

^{1,2}Program Studi Teknik Elektro Fakultas Teknik Universitas Balikpapan, Jln. Pupuk Raya Gn. Bahagia Balikpapan 76114 INDONESIA
(email: a_asni_b@yahoo.com, mayda.warunik@gmail.com)

secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan offset dan menghasilkan perubahan awal yang besar.

2.2.1 Metode Tuning Osilasi

Dalam metode tuning osilasi, mula-mula yang dilakukan adalah membuat $T_i = 0$ dan $T_d = 0$. Nilai K_p ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan. Dari keluaran yang beresilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan.



Gambar 2. 3 Osilasi Berkesinambungan dengan Periode Pcr

Setelah diperoleh nilai K_{cr} dan P_{cr} , dapat ditentukan nilai K_p , T_i dan T_d menggunakan metode Ziegler-Nichols berdasarkan pada Tabel 2.1[3].

Tabel 2.1 Aturan Dasar Ziegler-Nichols

Tipe Kendali	K_p	T_i	T_d
P	$0.5 \times K_{cr}$	-	-
PI	$0.45 \times K_{cr}$	$P_{cr} / 1.2$	-
PID	$0.6 \times K_{cr}$	$0.5 \times P_{cr}$	$0.125 \times P_{cr}$

2.3 Algoritme Recursive Least Square(RLS)

Algoritme adaptif Recursive Least Square (RLS) dapat dijelaskan oleh persamaan berikut:

$$(2.1)$$

dimana :

$$(2.2)$$

Forgetting factor (λ), pada algortime RLS, merupakan faktor yang secara eksponensial memberikan bobot yang lebih kecil kepada sampel-sampel error yang lebih lama dengan rentang nilai $0 < \lambda \leq 1$. Forgetting factor ini berguna untuk memastikan bahwa semakin jauh jarak data yang sebelumnya dengan data yang sedang diamati maka data tersebut semakin "dilupakan". Pada persoalan di atas nilai $\lambda = 1$ seperti yang tertera pada akhir listing program[4].

III. METODE PENELITIAN

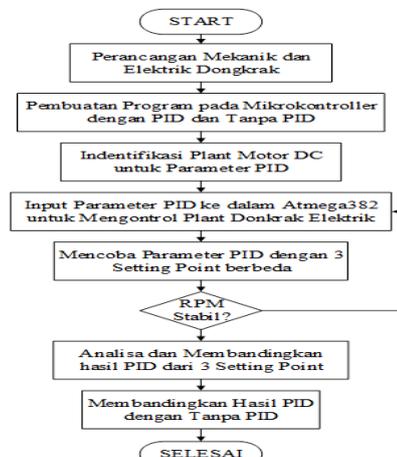
3.1 Peralatan Penelitian

Proses penelitian membutuhkan beberapa alat dan bahan yang akan digunakan untuk proses penelitian. MATLAB versi 7.11.0.0584 (R2010b) dengan memanfaatkan fasilitas tools simulink untuk menentukan nilai K_p , K_i , K_d dengan perangkat keras berupa Laptop dengan spesifikasi prosesor AMD E 450 ,G1.65GHz, dan RAM 2.00 GB yang juga akan digunakan untuk memprogram mikrokontroller Arduino UNO R3 sebagai

pengendali dan menampilkan data putaran mesin dan menyimpan data penelitian. Dongkrak Ulir dan Motor DC.

3.2 Prosedur Penelitian

Prosedur penelitian dilakukan dengan tujuan dalam penelitian ini dapat tercapai. Berikut ini merupakan diagram alir prosedur penelitian yang dirancang antara lain:



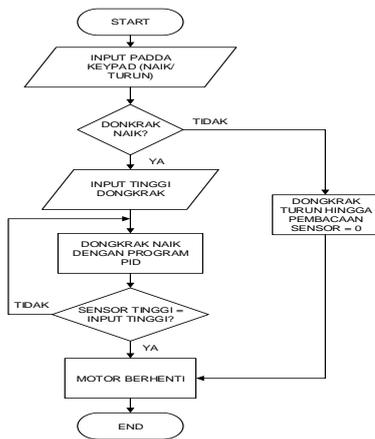
Gambar 3.1 Diagram Alir Proses Penelitian

Perancangan alat dongkrak menggunakan dongkrak ulir. Motor DC ditempatkan disisi sebagai penggerak *sprocket* yang kemudian di *couple* dengan *sprocket* yang berfungsi memutar poros dongkrak.

Pada perancangan elektrik terdapat beberapa proses yang perlu dilakukan seperti perancangan driver sensor optocoupler, perancangan driver motor, perancangan keypad, perancangan rangkaian LCD, dan perancangan rangkaian mikrokontroler arduino dan pemrogramannya.

Sistem utama dari perancangan alat ini terletak pada rangkaian mikrokontroller yang merupakan otak sistem. Rangkaian mikrokontroller ini berguna untuk memberikan sinyal saat alat dioperasikan. Agar mikrokontroller dapat bekerja maka memerlukan sinyal *clock* dari sebuah Kristal sebesar 16 MHz, yang terhubung dengan kapasitor 22 pF. Pada Arduino terdapat mikrokontroller Atmega328 yang di-*install* program untuk menggerakkan dongkrak dengan menggunakan *PID Controller*. Dalam perancangan *software* pada mikrokontroller Atmega328 berfungsi untuk mengatur arah, posisi dan respon motor dongkrak berdasarkan masukan kecepatan motor dan posisi dari motor.

Penulisan alur program pada Arduino IDE akan digambarkan pada diagram alir Gambar 3.2, antara lain:

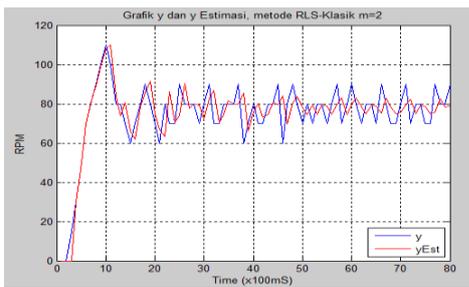


Gambar 3.2. Diagram Alir Program

IV. HASIL DAN ANALISIS

4.1 Perencanaan Respon Motor

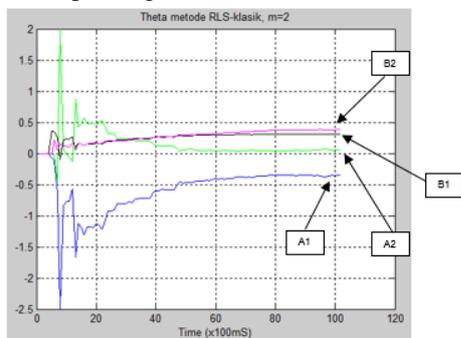
Setelah mendapatkan hasil identifikasi langkah selanjutnya adalah menggunakan metode RLS (*Recursive Least Square*), metode ini digunakan untuk mendapatkan respon yang lebih baik. Sehingga dari RLS akan mendapatkan grafik Y(t) dan yEst seperti ditunjukkan pada gambar berikut :



Gambar 4.1 Grafik Y(t) dan yEst

Dari gambar diatas merupakan gambar dari grafik Y(t) dan yEst setelah menggunakan metode RLS untuk mendapatkan hasil grafik yEst, dimana grafik yEst akan digunakan untuk mencari theta estimasi dari plan.

Berikut merupakan gambar hasil theta estimasi, antara lain:



Gambar 4.2 Grafik Theta Estimasi

Dari Gambar 4.2 di atas, didapatkan sebuah hasil grafik theta estimasi yang memiliki nilai akhir grafik mengalami kestabilan, sehingga dapat diambil nilai akhir tersebut, sebagai berikut:

$$A1 = -0,3425, \quad A2 = 0.0584$$

$$B1 = 0.3108, \quad B2 = 0.3926$$

Dengan nilai tersebut kemudian dapat digunakan sebagai perencanaan respon yang lebih baik. Selanjutnya dapat ditentukan fungsi alih dari respon tersebut berdasarkan persamaan 4.1, dengan urutan rumus sebagai berikut:

$$\text{_____} \quad 4.1(a)$$

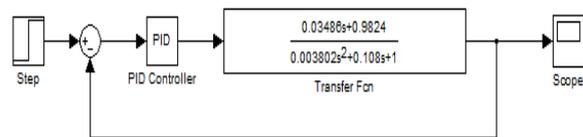
$$\text{_____} \quad 4.1(b)$$

$$\text{_____} \quad 4.1(c)$$

_____ bentuk fungsi transfer

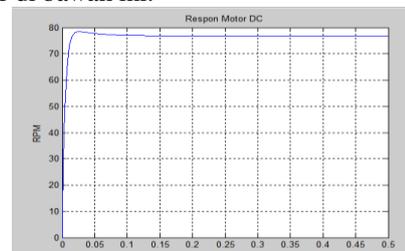
plant

Untuk mengetahui respon dari fungsi transfer dari *plant* tersebut, kemudian dilakukan pengujian pada simulink menggunakan fungsi transfer di atas,



Gambar 4.3 Simulasi Pada Simulink

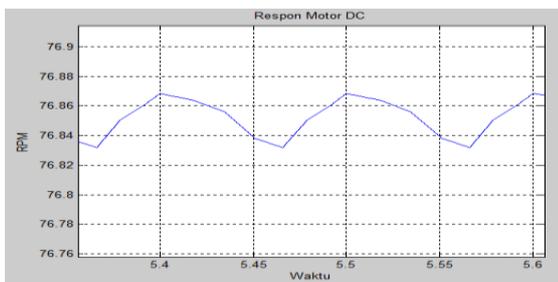
Sehingga dihasilkan respon pada *scope* seperti ditampilkan pada gambar di bawah ini.



Gambar 4.4 Grafik Fungsi Transfer Motor

4.3 Parameter Kontrol PID

Untuk mendapatkan parameter kontrol PID yang tepat dalam hal ini digunakan untuk menstabilkan putaran motor, dilakukan penerapan salah satu metode tuning osilasi *Ziegler-Nichols*. Metode ini digunakan karena saat dilakukan identifikasi respon motor diperoleh bentuk respon yang berupa osilasi berkesinambungan. Berikut grafik plan yang beresilasi secara kesinambungan setelah dilakukan *tuning* osilasi, antara lain:



Gambar 4.5 Grafik Beroperasi Secara Kesinambungan

Setelah dilakukan tuning berdasarkan fungsi transfer perencanaan respon motor, hingga menimbulkan grafik yang beroperasi secara kesinambungan, maka didapatkan nilai sebagai berikut: $K_{cr} = 25$ dan $P_{cr} = 0,1$

Nilai K_{cr} dan P_{cr} tersebut, selanjutnya diolah untuk mendapatkan nilai parameter K_p , K_i , dan K_d . Berdasarkan metode tuning *Ziegler-Nichols* yang dibahas pada bab 2, maka didapatkan nilai dari masing-masing parameter tersebut seperti yang ditunjukkan dalam tabel di bawah ini.

Tabel 4.1 Nilai Parameter Pengendali

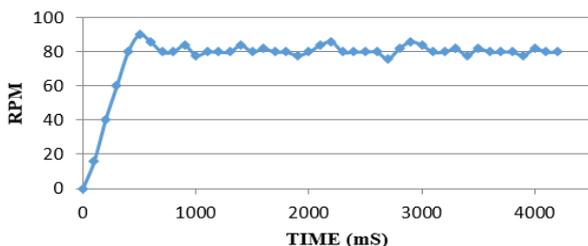
	K_p	T_i	T_d
P	12.5	-	-
PI	11.25	5,4	-
PID	15	0,05	0,0125

Dari tabel diatas memperoleh hasil parameter K_p kontrol adalah 15, sedangkan parameter K_i kontrol adalah 300 hasil dari pembagian nilai K_p terhadap T_i . Kemudian untuk mendapatkan nilai K_d kontrol adalah 0.1875 hasil kali dari K_p terhadap T_d .

4.4 Pengujian Parameter Kontrol

Sebelum penerapan parameter pada motor DC, dilakukan pengujian parameter kontrol PID dengan nilai parameter pengendali P bernilai 15, I bernilai 300 dan D bernilai 0.1875, kemudian hasil keluaran respon kontrol seperti Gambar 4.6 di bawah ini:

Hasil Respon Motor



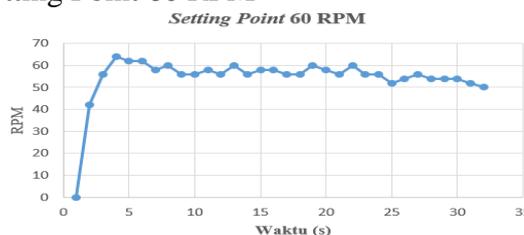
Gambar 4.6 Grafik Hasil Respon Motor DC Dengan Metode PID Setting Point 80

4.5 Pengujian Motor Terkontrol

Dalam pengujian respon motor dilakukan penerapan kontrol secara loop tertutup, dengan terpasangnya pengendali PID yasking tersusun secara paralel, pengolahan pengendali

PID tersebut menggunakan mikrokontroler yang telah diprogram dengan refrensi tertentu.

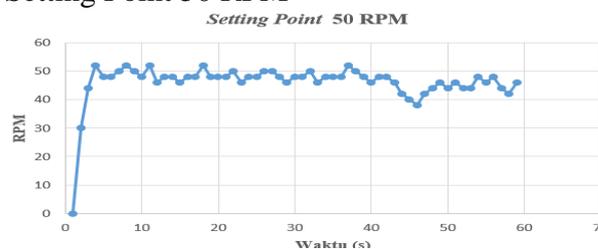
• Setting Point 60 RPM



Gambar 4.7 Grafik Keluaran Respon dengan Referensi 60 RPM

Dari Gambar 4.7 pengujian di atas didapatkan rata-rata kecepatan putaran motor tersebut sebelum terkena beban yaitu 57 RPM dan terdapat kecepatan tertinggi yaitu 64 RPM ketika awal dan 62 RPM ketika *steady state*. Ketika terkena beban, kecepatan rata-rata motor akan berkurang menjadi 55 RPM dengan 50 RPM sebagai kecepatan terendah dan 60 RPM sebagai kecepatan tertingginya.

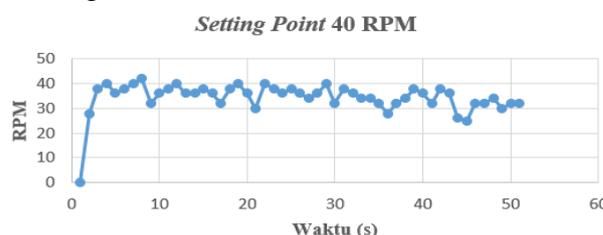
• Setting Point 50 RPM



Gambar 4.8 Grafik Keluaran Respon dengan Referensi 50 RPM

Dari Gambar 4.8 pengujian di atas didapatkan rata-rata kecepatan putaran motor tersebut sebelum terkena beban yaitu 48 RPM dan terdapat kecepatan tertinggi yaitu 52 RPM ketika awal dan 52 RPM ketika *steady state*. Ketika terkena beban, kecepatan rata-rata motor akan berkurang menjadi 45 RPM dengan 38 RPM sebagai kecepatan terendah dan 48 RPM sebagai kecepatan tertingginya.

• Setting Point 40 RPM



Gambar 4.9 Grafik Keluaran Respon dengan Referensi 40 RPM

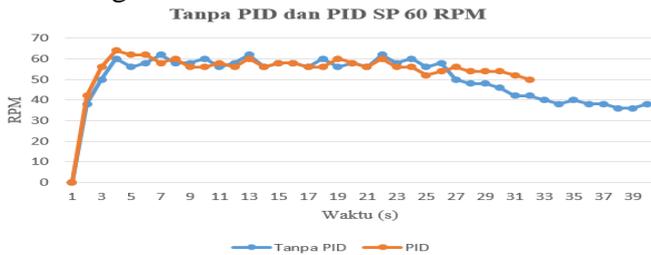
Dari Gambar 4.9 pengujian di atas didapatkan rata-rata kecepatan putaran motor tersebut sebelum terkena beban yaitu

36 RPM dan terdapat kecepatan tertinggi yaitu 42 RPM dan kecepatan terendah mencapai 30 RPM hingga mencapai titik *steady state*. Ketika terkena beban, kecepatan rata-rata motor akan berkurang menjadi 33 RPM dengan 25 RPM sebagai kecepatan terendah dan 38 RPM sebagai kecepatan tertingginya.

4.6 Hasil Motor Terkontrol dan Tidak Terkontrol

Setelah hasil yang telah dikontrol didapat maka analisa selanjutnya adalah membandingkan salah satu *setting point* kontrol dengan tanpa kontrol apabila sebelum dibebani sampai setelah dibebani.

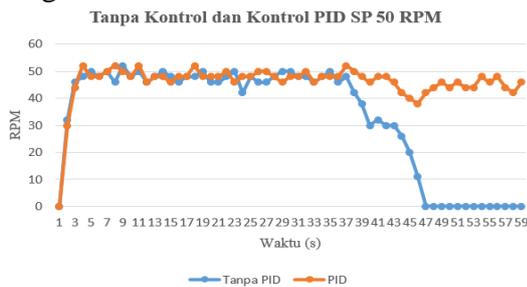
• Setting Point 60 RPM



Gambar 4.10 Grafik Perbandingan Tanpa Kontrol dan Kontrol PID SP 60 RPM

Pada Gambar 4.10 diatas terdapat perbandingan putaran ketika dibebani pada setting point 60 RPM, dimana motor dengan kontrol PID akan *steady state* selama 4 detik dari awal motor diberikan PWM. Sedangkan tanpa kontrol motor akan *steady state* membutuhkan waktu 7 detik dari awal diberi PWM. Ketika dongkrak terkena beban putaran motor dongkrak dengan kontrol dan tanpa kontrol sama-sama akan melambat sekitar 54 RPM dan 50 RPM, tetapi motor dengan kontrol akan naik kembali sesuai dengan setting point kontrol, sedangkan motor dongkrak tanpa kontrol putaran motor dongkrak tidak akan meningkat kembali.

• Setting Point 50 RPM

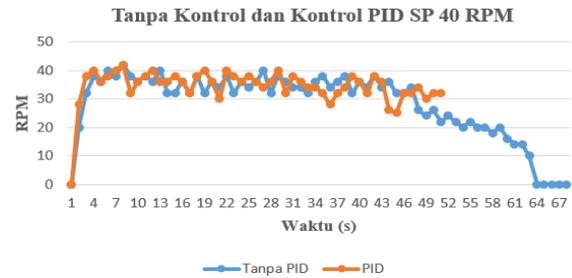


Gambar 4.11 Grafik Perbandingan Tanpa Kontrol dan Kontrol PID SP 50 RPM

Pada Gambar 4.11 diatas terdapat perbandingan putaran ketika dibebani pada *setting point* 50 RPM, dimana motor dengan kontrol PID akan *steady state* selama 5 detik dari awal motor diberikan PWM. Sedangkan tanpa kontrol motor akan *steady state* membutuhkan waktu 7 detik dari awal diberi PWM. Ketika dongkrak terkena beban, putaran motor

dongkrak dengan kontrol dan tanpa kontrol sama-sama akan melambat sekitar 48 RPM dan 38 RPM, tetapi motor dengan kontrol akan naik kembali sesuai dengan *setting point* kontrol, sedangkan motor dongkrak tanpa kontrol putaran motor dongkrak dengan *setting point* 50 RPM kecepatan putaran motor dongkrak *drop* secara cepat, sehingga mengakibatkan motor menjadi mati dan dongkrak tidak kuat mengangkat dengan pengaturan 50 RPM tanpa kontrol.

• Setting Point 40 RPM



Gambar 4.12 Grafik Perbandingan Tanpa Kontrol dan Kontrol PID SP 40 RPM

Pada Gambar 4.12 diatas terdapat perbandingan putaran ketika dibebani pada *setting point* 40 RPM, dimana motor dengan kontrol PID akan *steady state* selama 3 detik dari awal motor diberikan PWM. Sedangkan tanpa kontrol motor akan *steady state* membutuhkan waktu 5 detik dari awal diberi PWM. Ketika dongkrak terkena beban, putaran motor dongkrak dengan kontrol dan tanpa kontrol sama-sama akan melambat sekitar 36 RPM dan 26 RPM, tetapi motor dengan kontrol akan naik kembali sesuai dengan *setting point* kontrol, sedangkan motor dongkrak tanpa kontrol putaran motor dongkrak dengan *setting point* 40 RPM kecepatan putaran motor dongkrak *drop* secara cepat, sehingga mengakibatkan motor menjadi mati dan dongkrak tidak kuat mengangkat dengan pengaturan 40 RPM tanpa kontrol.

V. PENUTUP

5.1 Kesimpulan

Setelah dilakukan uji coba dongkrak elektrik dengan pengendali PID (*Proportional Integral Derivative*) pada bab sebelumnya maka dapat ditarik berbagai kesimpulan sebagai berikut:

- Mengatur kecepatan putaran motor dongkrak agar *steady state* menggunakan metode kontrol PID dengan *plant* putaran motor dongkrak, sehingga putaran motor dongkrak akan terkontrol sesuai dengan *setting point* kontrol.
- Setelah melakukan identifikasi *plant* yang kemudian di masukkan ke program RLS pada Matlab untuk mencari fungsi transfer motor. Setelah mendapatkan fungsi transfer motor proses selanjutnya mencari nilai control PID orde 2 dilakukan di program Simulink pada Matlab dan mendapatkan nilai Kcr 25 dan Pcr 0,1 sehingga

dengan menggunakan tabel parameter kontrol orde 2 mendapatkan nilai K_p 15 nilai T_i 0.05 dan nilai T_d 0,0125.

- Rata-rata kecepatan putaran motor dongkrak dengan kontrol PID akan dibagi menjadi 3 *setting point* yaitu 60, 50 dan 40. Ketika *setting point* 60 rata-rata kecepatan putaran motor dongkrak menjadi 56 dengan rata-rata 4 RPM dibawah *setting point*. Kemudian *setting point* 50 rata-rata kecepatan putaran motor dongkrak menjadi 47 dengan rata-rata 3 RPM dibawah *setting point*. Terakhir *setting point* 40 rata-rata kecepatan putaran motor dongkrak menjadi 37 dengan rata-rata 3 RPM dibawah *setting point*.

5.2 Saran

Saran yang dapat diberikan untuk perbaikan dan pengembangan simulasi ini adalah sebagai berikut :

1. Diharapkan adanya perbaikan terhadap mekanik pada *plant*.
2. Membandingkan hasil putaran dongkrak dengan pengendali jenis lain.

3. Diperlukan adanya pengawasan tentang keamanan pengangkatan setiap dongkrak bekerja.
4. Diperlukan pengembangan kontrol dengan nilai parameter K_p , T_i , T_d yang paling tepat.

REFERENSI

- [1] A. Su, N. Tanti, and T. G. Pandoyo, "Perencanaan Gearbox dan Perhitungan Daya Motor pada Modifikasi Dongkrak Ulir Mekanis Menjadi Dongkrak Ulir Elektrik," vol. 4, no. September, 2013.
- [2] O. Katsuhiko, *Modern Control Engineering*, Fifth. United States of America: Prentice-Hall, Inc, 2010.
- [3] "Metoda Tuning Ziegler-Nichols," 2011. .
- [4] Hugeng, E. Setyaningsih, and W. Meirista, "Pengurangan Noise Sepeda Motor dan Mesin Diesel dari Sinyal Bicara dengan Algoritma Recursive Least Square," Jakarta, 2013.