

## RANCANG BANGUN SISTEM KONTROL PID UNTUK PENGENDALIAN KECEPATAN PROTOTIPE LIFT BERBASIS LABVIEW

**Andrian Kusuma Solihin**

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia

e-mail : andriansolihin@mhs.unesa.ac.id

**Endryansyah**

Dosen Teknik Elektro, Fakultas Teknik, Universitas negeri Surabaya, Ketintang 60231, Indonesia

e-mail : syahryanend@unesa.ac.id

### Abstrak

Lift adalah angkutan transportasi vertikal digunakan untuk memindahkan barang. Untuk minimarket atau home industry yang memiliki struktur bangunan 2 sampai 3 lantai, banyak dijumpai masih menggunakan pengkataloran secara manual oleh tenaga manusia untuk memindahkan barang, Hal ini cukup tidak efisien. Tujuan dari penelitian ini adalah merancang dan mengatur kecepatan motor DC pada rancang bangun prototype Lift Barang menggunakan kontroler PID. Menggunakan metode Tyreus-Luyben terhadap hasil respon plant dengan menampilkan respon pada LabView 2014. Hasil dari penelitian menunjukkan bahwa Nilai parameter PID yang ditemukan dengan menggunakan metode Tyreus-Luyben yaitu didapatkan nilai  $K_p = 4.14$ ,  $K_i = 1,727$ , dan  $K_d = 0.042$ . Kemudian nilai tersebut diterapkan pada sistem dan memberikan respon yang baik pada setpoint 30 Rpm dengan ( $t_d$ ) sebesar 1,381 detik, nilai waktu naik ( $t_r=5\%-95\%$ ) sebesar 5,868 detik, nilai waktu naik ( $t_r=10\%-90\%$ ) sebesar 4,379 detik , nilai waktu tunak ( $t_s=5\%$ ) sebesar 5.979 detik, nilai waktu tunak ( $t_s=2\%$ ) sebesar 7,972 detik, nilai waktu tunak ( $t_s=0,5\%$ ) sebesar 9,965 detik dan nilai *error steady state* sebesar 1.59%.

**Kata Kunci:** Prototipe Lift, Motor DC, Kontroler PID, *LabView* 2014

### Abstract

Lifts are vertical transports used to move goods. For minimarkets or home industries that have 2 to 3-storey building structures, many are still found using manual labor by human labor to move things, this is quite inefficient. The purpose of this study is to design and regulate the speed of a DC motor in the design of a prototype elevator using a PID controller. Using the Tyreus-Luyben method to the results of the plant response by displaying responses in LabView 2014. The results of the study showed that the PID parameter values found using the Tyreus-Luyben method were obtained values of  $K_p = 4.14$ ,  $K_i = 1,727$ , and  $K_d = 0.042$ . Then the value is applied to the system and gives a good response at a setpoint of 30 Rpm with ( $t_d$ ) of 1,381 seconds, the time value rises ( $t_r = 5\% -95\%$ ) of 5,868 seconds, the value of time rises ( $t_r = 10\% -90\%$ ) of 4,379 seconds, the value of steady time ( $t_s = 5\%$ ) is 5 ,979 seconds, steady time value ( $t_s = 2\%$ ) of 7.972 seconds, steady time value ( $t_s = 0.5\%$ ) of 9.965 seconds and steady state error value of 1.59%.

**Keywords:** Prototype Lift, DC Motor, PID Controller, LabView 2014

### PENDAHULUAN

Pada zaman sekarang ini perkembangan teknologi sangat pesat. Salah satu teknologi yang sering kita jumpai di gedung bertingkat yaitu pada alat transportasi untuk menaikkan dan menurunkan muatan. Kita sering menyebutnya

dengan lift. penelitian dilakukan oleh Andi Adriansyah dengan judul rancang bangun prototipe *elevator* menggunakan *microcontroller* Arduino ATMEGA 328P (Adriansyah, 2013). Pada penelitian ini, penulis merancang sebuah prototipe lift 3 lantai menggunakan motor DC sebagai katrol pengangkat

dan sebuah arduino ATMEGA 328P sebagai *controllernya*.

Pada penelitian ini sistem lift menggunakan controller PID. Dengan mengurangi sinyal kesalahan yang terjadi saat sistem bekerja, serta mampu memberikan keluaran sinyal kontrol yang memiliki respon cepat, *error steady state* kecil, meminimalisir terjadinya *overshoot*. Semakin kecil nilai *error* yang terjadi, maka semakin baik kinerja sistem kontrol yang diterapkan. Kontroler ini memiliki nilai parameter proporsional sebagai nilai pengali *error* untuk nilai koreksi. Nilai parameter integral sebagai perbaikan kesalahan *steady state* mencapai nol. Dan nilai parameter *deferensial* sebagai perbaikan respon *transien* dan meredam *osilasi*. Dari tinjauan tersebut, maka penulis melakukan penelitian ini dengan judul "Rancang Bangun Sistem Kontrol PID Untuk Pengendalian Kecepatan Prototipe Lift Berbasis *LabView*".

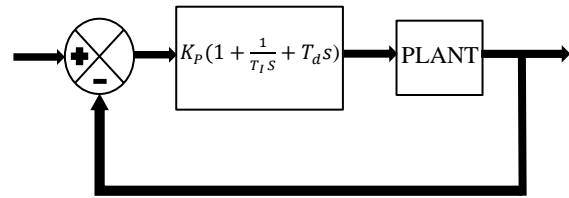
**KAJIAN PUSTAKA**  
**Analisis Sistem Lift**

Sistem katrol Lift ini digerakkan oleh penggerak motor DC yang terhubung dengan Kereta (katrol). Motor bergerak dalam keadaan berbeban atau tidak berbeban. Motor ini selalu memutar beban dilengkapi dengan sistem transmisi *Geared* ataupun transmisi *Gearless*. Dengan menggunakan mesin arus searah yang memiliki *torsi* dapat diatur dengan mudah. Sebagai langkah awal, kita perlu membuat model matematika yang akan digunakan untuk mempelajari sifat-sifat sistem dan mengontrolnya agar berjalan sesuai kecepatan yang diinginkan.

**PID (Proportional-Integral-Derivative)**

PID (*Proportional - Integral - Derrivative*) adalah salah satu metode kontrol klasik yang sering digunakan saat ini terutama pada industri karena struktur sederhana dan bekerja stabil (Alfa Syahroni. 2018).

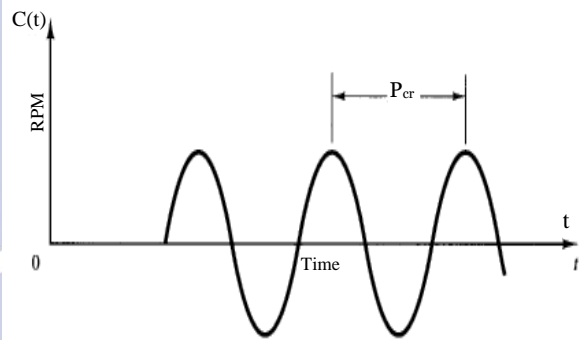
PID memiliki dinamika kontrol yang optimal dengan *error steady state* sistemnya bernilai nol, respon sistem yang cepat (*Rise time* pendek), tidak ada *osilasi* dan *stabilitas* yang lebih tinggi. Perlunya menggunakan komponen *gain* turunan selain pengontrol PI adalah untuk menghilangkan *overshoot* dan *osilasi* yang terjadi dalam respons *output* sistem. Salah satu keuntungan utama dari pengontrol PID adalah dapat digunakan dengan proses urutan yang lebih tinggi termasuk lebih dari penyimpanan energi tunggal (Temel. 2013). Berikut ini merupakan diagram blok sistem kontrol PID dengan sebuah *input* dan proses .



**Gambar 1.** Diagram Blok Sistem Kontrol PID (Sumber: Ogata. 2010)

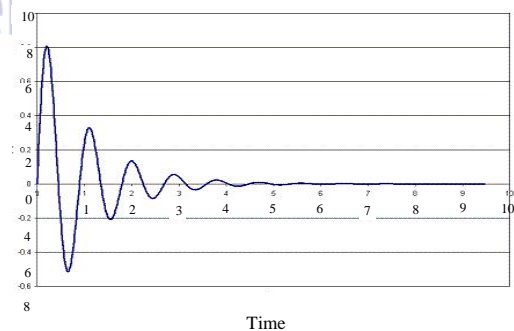
**Metode Tuning Tyreus Luyben**

Prosedur metode *Tyreus - Luyben* (TL) hampir sama dengan metode *Ziegler-Nichols* tetapi terdapat perbedaan pada *setting* akhir parameter kendali. Metode ini dilakukan dengan menggunakan *proportional band* saja. Nilai *Kp* dinaikkan dari 0 hingga tercapai nilai *Kp* yang menghasilkan *osilasi* yang konsisten. Maksud dari *osilasi* konsisten bisa dilihat pada Gambar 2. Nilai *Kp* yang menyebabkan *osilasi maksimum* ini disebut sebagai *critical gain* (*Kcr*), sedangkan waktu yang dibutuhkan antar puncak *osilasi* disebut sebagai periode kritis (*Pcr*).



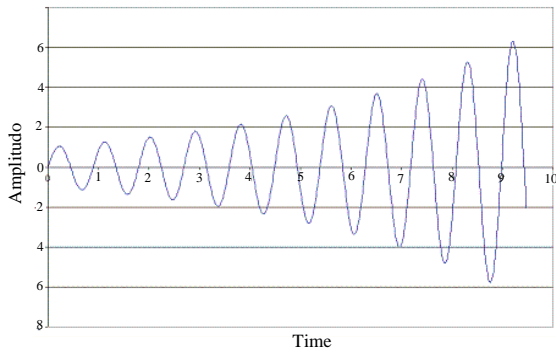
**Gambar 2.** Osilasi Konsisten (Sumber: Ogata. 2010)

Jika *Kp* ini terlalu kecil, sinyal *output* akan teredam mencapai nilai titik keseimbangan setelah ada gangguan, seperti terlihat pada gambar di bawah ini.



**Gambar 3.** Sistem Teredam (Sumber: Ogata. 2010)

Sebaliknya, jika *Kp* terlalu besar, *osilasinya* akan tidak stabil dan membesar, seperti Gambar 4.



**Gambar 4.** Sistem Tidak Tereadam  
(Sumber: Ogata. 2010)

Jika dengan metode ini tidak diperoleh *osilasi* yang konsisten, maka metode ini tidak dapat dilakukan. Dari metode ini akan diperoleh nilai *critical gain*  $K_{cr}$  dan periode kritis  $P_{cr}$ , seperti pada Gambar 2.

**Tabel 1.** Penentuan Parameter PID Tyreus-Luyben  
(Sumber: Taan, Ahmad. 2015)

Type of Controller	$K_p$	$T_i$	$T_d$
PI	$0.45K_{cr}$	$2.2P_{cr}$	0
PID	$0.313K_{cr}$	$2.2P_{cr}$	$0,16P_{cr}$

Metode ini dapat digunakan untuk kendali dengan kombinasi PI dan PID dengan sistem *close loop* yang memiliki umpan balik sebagai nilai parameter yang akan di tentukan (Taan. 2015). Dengan memasukkan nilai  $K_{cr}$  dan  $P_{cr}$  yang telah didapatkan pada grafik simulasi *plant*.

### Rotary Encoder



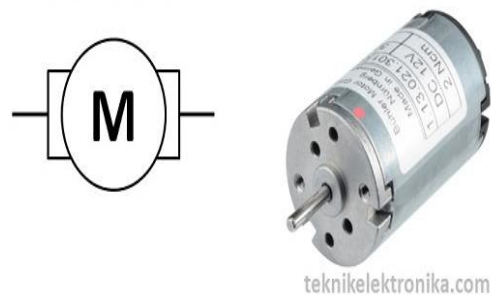
**Gambar 5.** Rotary Encoder  
(Sumber: Erdon.2012)

*Rotary incremental encoder* jenis omron dengan tegangan 12 Volt DC. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary incremental encoder* tersebut. Pada *Rotary Encoder* terdapat LED dimana ditempatkan disalah satu sisi piringan yang kemudian cahaya akan menuju piringan, disisi lainnya terdapat *photo-transistor* sehingga *photo-transistor* ini dapat mendeteksi cahaya yang ada

dipancarkan oleh LED. Dimana piringan tipis dihubungkan dengan poros motor, atau perangkat berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan mengakibatkan cahaya dari LED dapat mencapai *photo-transistor* melalui lubang-lubang yang ada, maka *photo-transistor* akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi.. (Anonymous, 2016).

### Motor DC

Motor DC memerlukan *suplai* tegangan yang searah pada kumparan medan untuk diubah menjadi energi *mekanik* (Nugraha, 2015). Pada tugas akhir ini motor yang digunakan untuk menggerakkan sitem *prototype* lift adalah motor DC magnet permanen dengan tegangan *maksimum* 24 V dan memiliki *torsi* hingga 25 Kg\*cm.



**Gambar 6.** Simbol dan Bentuk Motor DC  
(Sumber: Kho. 2017)

Terdapat dua bagian utama pada sebuah Motor Listrik DC, yaitu *stator* dan *rotor*.

1. *Stator* adalah bagian motor yang tidak berputar, bagian yang statis ini terdiri dari rangka dan kumparan medan.
2. *Rotor* adalah bagian yang berputar, bagian *Rotor* ini terdiri dari kumparan *jangkar*.

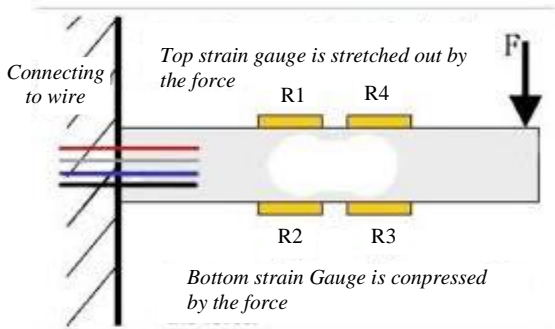
Dua bagian utama ini dapat dibagi lagi menjadi beberapa komponen penting yaitu diantaranya adalah *Yoke* (kerangka magnet), *Poles* (kutub motor), *Field winding* (kumparan medan magnet), *Armature Winding* (kumparan jangkar), *Commutator* (komutator) dan *Brushes* (kuas/sikat arang).

### Loadcell

*Load cell* atau sel beban adalah *sensor* yang terdiri dari sebuah strain gauge atau lebih, yang ditempelkan pada batang berbahan logam yang berbentuk cincin. Jumlah strain gauge dalam sebuah *load cell* bisa disesuaikan dengan kebutuhan. Strain gauge yang ada pada *load cell* terbuat dari bahan *foil grid*. *Foil grid*



sendiri merupakan kawat tipis berukuran panjang yang disusun secara zig-zag.

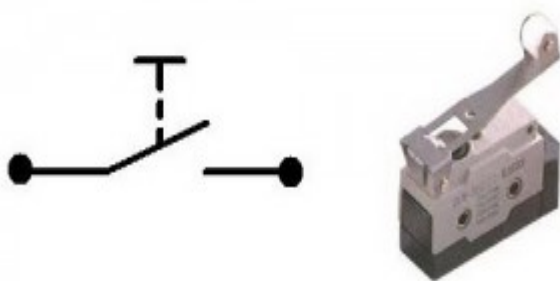


**Gambar 7. Loadcell**  
(Sumber: Khakim, 2015)

Cara kerja dari *strain gauge* ini adalah saat *strain gauge* mendapat tarikan maka akan menyebabkan perubahan panjang pada kawat tipis penyusunnya sehingga menyebabkan bertambahnya *resistansi* yang dihasilkan, perubahan *resistansi* inilah yang akhirnya dimanfaatkan sebagai patokan perubahan pada *sensor* beban (*Load Cell*). (Khakim, 2015).

**Limit Switch**

*Limit switch* termasuk dalam kategori *sensor* mekanis yaitu *sensor* yang akan memberikan perubahan elektrik saat terjadi perubahan *mekanik* pada *sensor* tersebut. Penerapan dari *limit switch* Sebagai penanda lift ketika talah mencapai di lantai yang di inginkan . Simbol *limit switch* ini menggunakan merek *Omron* type Z-15GW2 dengan batas tegangan 125 VDC - 250 VDC ditunjukkan pada Gambar 8.



**Gambar 8. Komponen Limit Switch**  
(Sumber: Purba, 2010)

**Analisis Karakteristik Orde Satu**

Dari model matematis sebuah sistem, *orde* dari suatu sistem dapat dilihat dari besar pangkat *variabel s* (dalam *transformasi Laplace*). Suatu sistem dikatakan ber-*orde* satu jika fungsi alihnya mempunyai *variabel s* dengan pangkat tertinggi satu. Bentuk fisisnya bisa berupa rangkaian listrik RC, sistem *termal*, atau sistem

lainnya (Ogata, 2010). Model sistem *orde* satu secara matematis dapat dituliskan sebagai berikut :

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1} \tag{1}$$

Keterangan :

$C(s)$  = Output sistem .

$R(s)$  = Input sistem.

$K$  = Gain Overall.

$\tau s + 1$  = Waktu yang dibutuhkan untuk mencapai 63,2% (detik) pada *orde* satu.

Untuk menentukan parameter  $K$  (*Gain Overall*) jika sistem *linier*, maka hubungan  $Y_{ss}$  dengan  $X_{ss}$  dapat dituliskan sebagai berikut :

$$K = \frac{Y_{ss}}{X_{ss}} \tag{2}$$

Keterangan :

$X_{ss}$  = Setpoint.

$Y_{ss}$  = Hasil Respon.

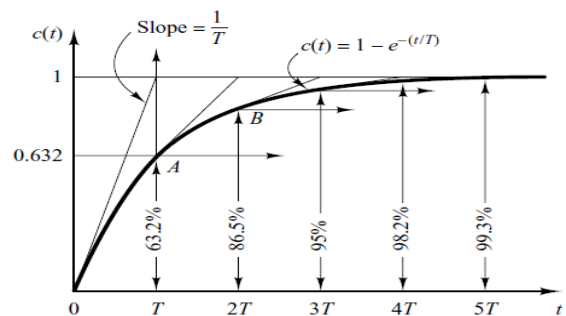
Untuk menentukan konstanta waktu ( $\tau$ ) dapat dihitung melalui respon keluaran sistem ketika mencapai 63,2% dari hasil akhirnya  $C(\tau)$  dapat dihitung sebagai berikut :

$$C(\tau) = 0.632 \times Y_{ss} \tag{3}$$

Keterangan :

$C(\tau)$  = Keluaran sistem ketika 63,2% dari hasil akhir.

Bentuk kurva tanggapan *eksponensial* orde satu ditunjukkan pada Gambar 8.



**Gambar 9. Kurva Tanggapan Eksponensial Orde 1**  
(Sumber : Ogata. 2010)

Rumus analitik untuk perhitungan parameter respon dari orde satu sebagai berikut :

1. Waktu tunda (td)

2. Waktu naik ( $t_r$ )
- Waktu naik (5%-95%)  
 $T_r = \tau \ln 19$
  - Waktu naik (10%-90%)  
 $T_r = \tau \ln 9$
3. Waktu Tunak ( $t_s$ )
- Waktu tunak (0.5%)  
 $T_s = 5\tau$
  - Waktu tunak (2%)  
 $T_s = 4\tau$
  - Waktu tunak (5%)  
 $T_s = 3\tau$
4. Persentasi *Error Steady State* ( $E_{ss}$ )

$$E_{ss} = \left| \frac{Y_{ss} - X_{ss}}{X_{ss}} \right| \times 100\% \quad (10)$$

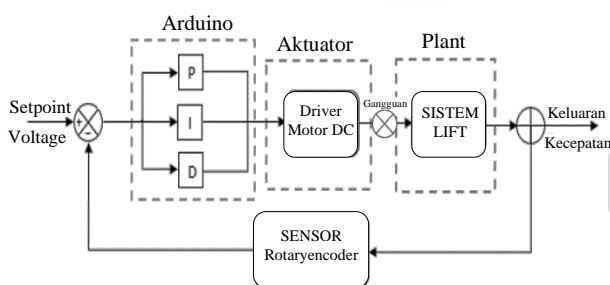
## METODE PENELITIAN

### Pendekatan Penelitian

Penelitian ini menggunakan pendekatan *kuantitatif*. dapat diartikan sebagai metode penelitian yang berlandaskan pada filsafat *positivisme*, digunakan untuk meneliti pada populasi atau sampel tertentu, pengumpulan data menggunakan instrument penelitian, *analisis* data bersifat *kuantitatif/statistik*, dengan tujuan untuk menguji *hipotesis* yang ditetapkan (Sarwono,2006).

### Desain Sistem

Desain sistem *PID* (*propotional integrative derevativ*) untuk pengendalian kecepatan motor DC pada *prototype* lift Barang ditunjukkan pada Gambar 10.



**Gambar 10.** Diagram Blok Sistem Lift  
(Sumber : Dokumen Pribadi, 2019)

Nilai masukan (*Setpoint*) sistem berupa tegangan. *Kontroler* PID akan diproses pada Arduino yang nantinya akan mengatur kecepatan pada lift. Hasil dari keluaran *kontroler* berupa sinyal PWM yang akan masuk ke *driver* motor. *Driver* motor akan mengatur arah putar dan kecepatan

motor DC. *Sensor rotaryencoder* akan bekerja untuk mengukur kecepatan motor DC setelah adanya beban yang telah diberikan pada lift dan kemudian menjadikan nilai pembacaan sebagai nilai umpan balik pada keluaran yang dihasilkan berupa kecepatan.

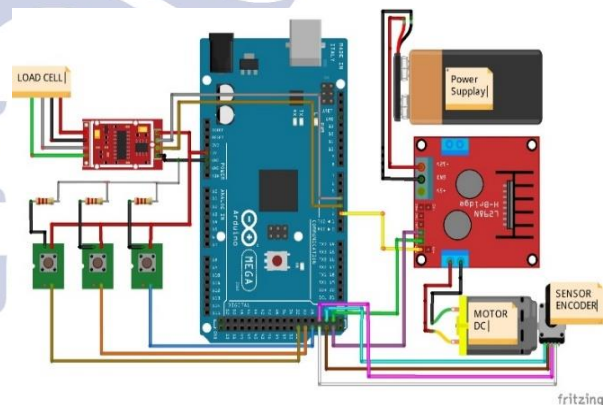
### Rancang bangun *Hardware*

Rancang bangun hardware *single link flexible joint manipulator* dapat dilihat pada Gambar 11.



**Gambar 11.** Kerangka Rancang Bangun *Plant* Lift  
(Sumber : Dokumen Pribadi, 2019)

Pada skripsi ini menggunakan *plant* Lift Barang sebagai tempat sistem bekerja, terdiri dari 3 lantai lift dan satu atap sebagai tempat pengontrolan lift. Berikut ini adalah merupakan model desain rancang bangun Lift dengan dimensi keseluruhan 30cm x 30cm x 175 cm.

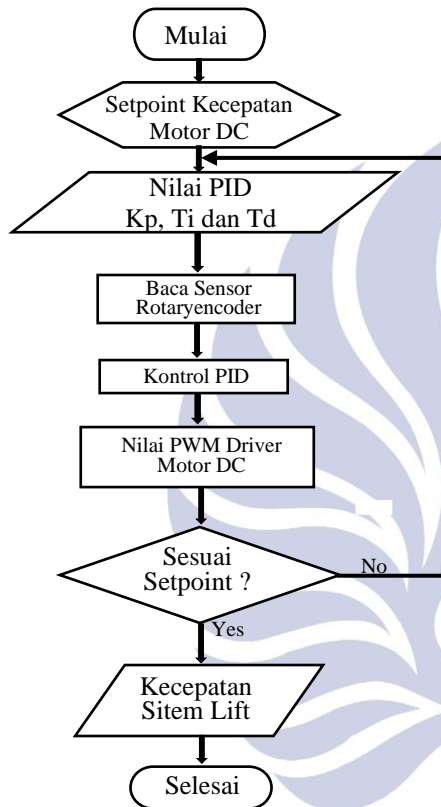


**Gambar 12.** Diagram Pengkabelan Sistem Lift.  
(Sumber : Dokumen Pribadi, 2019)

Perancangan hardware terdiri dari dua bagian Perancangan hardware merupakan proses desain bagian-bagian penyusun dari sistem pengaturan kecepatan motor DC. Bagian tersebut meliputi komponen yang digunakan dalam penelitian ini adalah Motor DC, *Rotary Encoder*, *Load Cell*, *Limit switch* dan *driver* motor DC.

**Rancang Bangun Software**

Rancang bangun *software* yang dimaksud adalah sebuah program yang akan diintegrasikan antara Arduino dengan *LabView*. Nilai PID diinisialisasikan terlebih dahulu dengan menentukan *parameter* Kp, Ti dan Td. Pada prototipe Sistem lift ini memberi masukan setpoint kecepatan yang telah di tentukan dengan mengintegrasikannya kedalam Arduino, kemudian *sensor rotay encoder* akan mengirimkan data kecepatan dari motor DC.



**Gambar 13.** Diagram Alir Software (Sumber : Dokumen Pribadi, 2019)

Setelah itu menghasilkan output berupa kecepatan pada sistem Prototipe lift yang sesuai dengan *setpoint*, kemudian melihat respon *dinamik* pada tampilan GUI *LabVIEW*.

**HASIL DAN PEMBAHASAN**

Pada Bab ini akan dibahas proses pemodelan dan identifikasi sistem, penentuan parameter PID, dan implementasi PID dan analisis terhadap respon *plant*.

**Pemodelan dan Identifikasi Sistem**

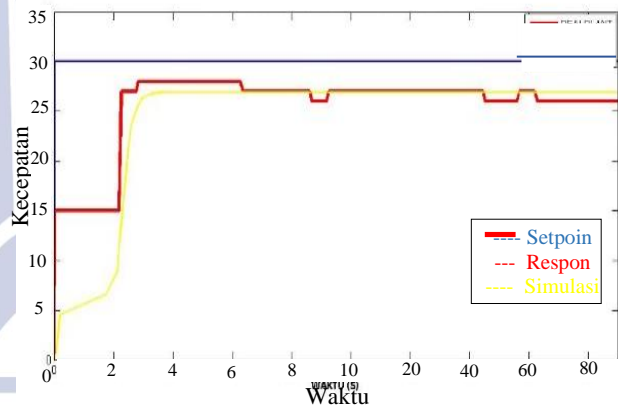
Hasil perancangan *software* kode Arduino terdiri dari kode sistem kecepatan lift .Sedangkan pada *software* Labview ditunjukkan kode *blok* dan *front panel*.



**Gambar 14.** Tampilan GUI *LabVIEW* (Sumber: Dokumen Pribadi. 2019)

Sebelum merancang parameter dari pengendali PID, dihitung parameter dari respon sistem riil plant secara open loop pada setpoint 30 Rpm dengan nilai  $\tau = 2,2014$  detik. Parameter respon plant riil dengan setpoint 30 Rpm secara open loop menggunakan persamaan (4), (5), (6), (7), (8), (9), dan(10).

$$\frac{C(s)}{R(s)} = \frac{0,897855}{2,2014s + 1}$$

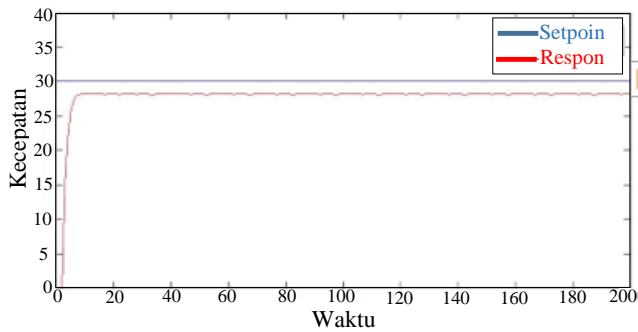


**Gambar 15.** Perbandingan Respon Sistem Riil dan Respon Sistem Simulasi tanpa Kontroller (Sumber: Dokumen Pribadi. 2019)

**Penentuan Parameter PID**

Setelah didapatkan fungsi alih *plant*, selanjutnya adalah menentukan nilai parameter PID yaitu nilai Kp, Ki, dan Kd sebelum diaplikasikan pada *plant*. Penentuan nilai parameter PID dilakukan dengan menaikkan nilai Kp hingga menghasilkan respon sistem yang berkelanjutan. Percobaan untuk mendapatkan hasil respon *osilasi* berkelanjutan dilakukan dengan 3 nilai Kp yaitu 11, 12, dan 13 . Berikut ini merupakan respon hasil dari simulasi menggunakan nilai Kp sebesar 11, 12, dan 13. Mendapatkan nilai Kcr 13 .





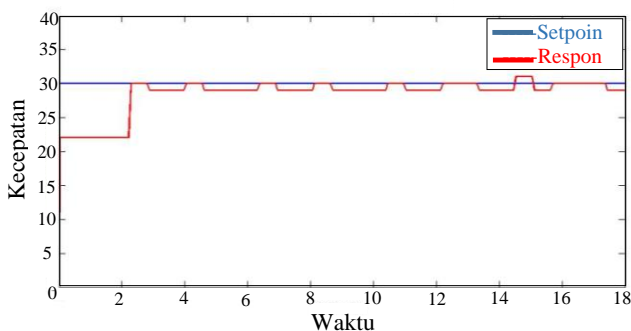
**Gambar 16.** Respon Simulasi Sistem dengan  $K_p = 13$   
(Sumber: Dokumen Pribadi. 2019)

Pada Gambar 16 menampilkan grafik respon sitem pada  $K_p$  13 sudah beresilasi dan telah di dapatkan antar puncak gelombang. Dalam metode ini, selain membutuhkan nilai  $K_{cr}$  untuk menentukan nilai kontroler *proporsional*, nilai dari kontroler *integral* dan *derivative* juga diperlukan dengan cara mencari nilai  $P_{cr}$ . Nilai  $P_{cr}$  merupakan waktu antar puncak *osilasi*. Puncak *osilasi* dari respon ini adalah 28. Berdasarkan data  $P_{cr}$  yang telah diambil secara acak tersebut maka nilai  $P_{cr}$  yang digunakan untuk menentukan nilai kontroler *integral* dan *derivative* adalah 0.263 s. Dengan nilai  $K_{cr} = 13$  dan  $P_{cr} = 0.263$  s. Sehingga nilai dari  $K_p$ ,  $K_i$ , dan  $K_d$  berdasarkan Tabel 1 adalah  $K_p = 4,069$ ,  $K_i = 1.727$ , dan  $K_d = 0.042$ .

### Implementasi PID dan Analisis Respon Plant

Setelah mendapatkan nilai parameter PID, nilai tersebut kemudian dimasukkan ke dalam program Arduino untuk diimplementasikan pada plant system dengan *Setpoint* yang ditentukan dan dilakukan pengujian terhadap gangguan berupa beban. *Implementasi* kendali PID pada plant sistem Lift dilakukan sebagai berikut.

#### 1. Implementasi PID *Setpoint* 30 Rpm Tanpa Beban



**Gambar 17.** Grafik Respon *Plant* dengan Kontroler PID *Setpoint* 30 Rpm Tanpa Beban  
(Sumber: Dokumen Pribadi. 2019)

Pada Gambar 17 berikut ini menampilkan grafik respon sistem menggunakan kontroler PID Tyreus-Luyben dengan *Setpoint* 30 rpm selama  $\pm 18$  detik. Berikut merupakan karakteristik responnya.

#### a. Time Constant ( $\tau$ )

Nilai  $C(\tau)$  dari kedua respon ini adalah 1.007s.

#### b. Settling Time

*Settling Time* merupakan waktu saat respon mengalami *steady state*. Dengan demikian *Settling Time* untuk metode Tyreus-Luyben adalah 5.979s.

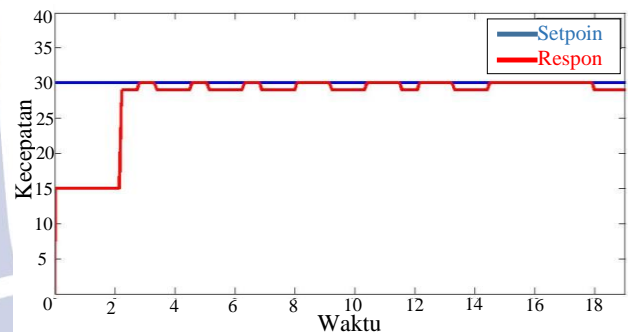
#### c. Rise Time

*Rise Time* (10%-90%) berdasarkan Persamaan 6 adalah  $\tau \times \ln 9$ . Dengan demikian *Rise Time* (10%-90%) untuk metode Tyreus-Luyben adalah 4,379s.

#### d. Delay Time

*Delay Time* berdasarkan Persamaan 4 adalah  $\tau \times \ln 2$ . Dengan demikian *Delay Time* untuk metode Tyreus-Luyben adalah 1.381s.

#### 2. Implementasi PID *Setpoint* 30 Rpm Beban 100gram



**Gambar 18.** Grafik Respon *Plant* Menggunakan Kontroler PID dengan Beban pada *Setpoint* 30 Rpm 100 Gram  
(Sumber: Dokumen Pribadi. 2019)

Pada Gambar 18 dibawah ini menampilkan grafik respon sistem menggunakan kontroler PID Tyreus-Luyben dengan *Setpoint* 30 rpm dengan beban 100gram selama  $\pm 27$  detik. Berikut merupakan karakteristik responnya.

#### a. Time Constant ( $\tau$ )

Nilai  $C(\tau)$  dari kedua respon ini adalah 1,86s.

#### b. Settling Time ( $\pm 0.5\%$ )

*Settling Time* merupakan waktu saat respon mengalami *steady state*. Dengan demikian, *Settling Time* untuk metode Tyreus-Luyben adalah 5.844s.

#### c. Rise Time

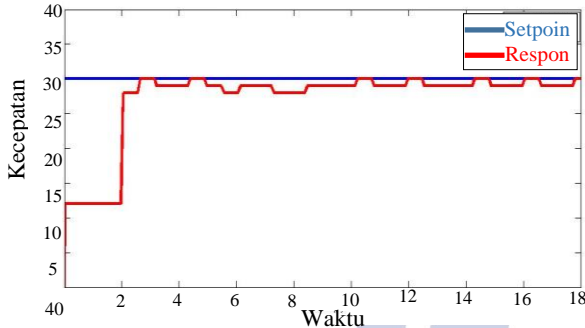
*Rise Time* (10%-90%) berdasarkan Persamaan 6 adalah  $\tau \times \ln 9$ . Dengan demikian, *Rise Time*

(10%-90%) untuk metode Tyreus-Luyben adalah 4,28s.

d. *Delay Time*

*Delay Time* berdasarkan Persamaan 4 adalah  $\tau \times \ln 2$ . Dengan demikian, *Delay Time* untuk metode Tyreus-Luyben adalah 1.35s.

3. Implementasi PID *Setpoint* 30 Rpm Beban 200 gram



**Gambar 19.** Grafik Respon Sistem Menggunakan Kontroler PID Tyreus Luyben dengan beban 200 gr (Sumber: Dokumen Pribadi. 2019)

Pada Gambar 19. menampilkan grafik respon sistem menggunakan kontroler PID Tyreus-Luyben dengan *Setpoint* 30rpm dengan beban 200 gram selama  $\pm 27$  detik. Berikut merupakan karakteristik responnya.

a. *Time Constant* ( $\tau$ )

Nilai  $C(\tau)$  dari kedua respon ini 2.7425s.

b. *Settling Time*

*Settling Time* merupakan waktu saat respon mengalami *steady state*. Dengan demikian, *Settling Time* untuk metode Tyreus-Luyben adalah 6.813s.

c. *Rise Time*

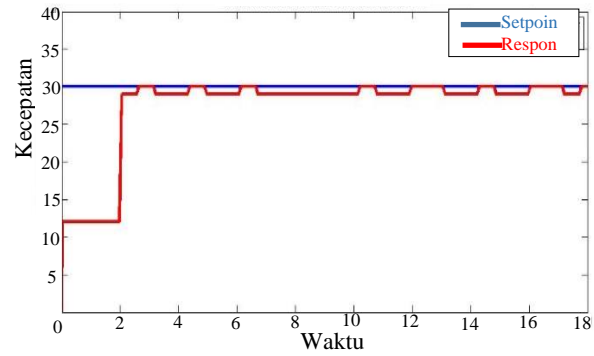
Nilai *Rise Time* (10%-90%) berdasarkan Persamaan 6 adalah  $\tau \times \ln 9$ . Dengan demikian, nilai dari *Rise Time* (10%-90%) untuk metode Tyreus-Luyben adalah 4.989s.

d. *Delay Time*

Nilai *Delay Time* berdasarkan Persamaan 4 adalah  $\tau \times \ln 2$ . Dengan demikian, nilai dari *Delay Time* untuk metode Tyreus-Luyben adalah 1.574s.

4. Implementasi PID *Setpoint* 30 Rpm Beban 300 gram

Pada Gambar 20. menampilkan grafik respon sistem menggunakan kontroler PID Tyreus-Luyben dengan *Setpoint* 30 rpm dengan beban 300 gram selama  $\pm 27$  detik. Berikut merupakan karakteristik responnya.



**Gambar 20.** Grafik Respon Sistem Menggunakan Kontroler PID Tyreus Luyben dengan *Setpoint* 30 rpm dengan beban 300 gram (Sumber: Dokumen Pribadi. 2019)

e. *Time Constant* ( $\tau$ )

Nilai  $C(\tau)$  dari kedua respon ini adalah 2.7425s.

f. *Settling Time*

*Settling Time* merupakan waktu saat respon mengalami *steady state*. Dengan demikian, *Settling Time* untuk metode Tyreus-Luyben adalah 6.633s.

g. *Rise Time*

Nilai *Rise Time* (10%-90%) berdasarkan Persamaan 6 adalah  $\tau \times \ln 9$ . Dengan demikian, nilai dari *Rise Time* (10%-90%) untuk metode Tyreus-Luyben adalah 4.85s.

h. *Delay Time*

Nilai *Delay Time* berdasarkan Persamaan 4 adalah  $\tau \times \ln 2$ . Dengan demikian, nilai dari *Delay Time* untuk metode Tyreus-Luyben adalah 1.532s.

Berdasarkan analisis terhadap hasil respon saat diberikan ketiga setpoint tersebut, Nilai-nilai variabel dari karakteristik respon transien dan steady state dapat dirangkum pada Tabel 2 berikut ini.

**Tabel 2.** Nilai Variabel terhadap Hasil Respon *Plant* (Sumber: Dokumen Pribadi. 2019)

Variabel	<i>Setpoint</i> 30 Rpm Pada Beban Berbeda			
	0	100	200	300
$\tau$ (s)	1,007	1,86	2,7425	2,7425
$T_s$ (s)	5.979 s	5.844 s	6.813 s	6.633 s
$T_r$ (s)	4,379 s	4,28 s	4,989 s	4,85 s
$T_d$ (s)	1,381s	1,35s	1,574s	1,532s
Ess (s)	1.59.%	1.785%	1.882%	2.075%



Berdasarkan data hasil yang telah diperoleh dan dilakukan analisis pada tabel 2 . Dapat diketahui bahwa semakin berat beban yang di berikan maka respon yang di hasilkan pada sitem lift akan mengalami kenaikan *error stady state*. Dengan error pada setiap beban sebesar 1,59%, 1,785%, 1,882% dan 2.075 %. Masih dalam keadaan normal dengan perubahan *error* yang tidak terlalu besar.

## PENUTUP

### Simpulan

Berdasarkan data Tabel 2 hasil yang telah diperoleh dan setelah dilakukan *analisis*, maka didapatkan kesimpulan sebagai berikut dengan model fungsih plant

$$\frac{C(s)}{R(s)} = \frac{0,897855}{2,2014s + 1}$$

Dengan mencari respon plant pada sistem lift didapatkan fungsi alih dengan *error* terkecil adalah pada *setpoint* 30 Rpm dengan nilai *error* rata-rata sebesar 10,214%. Nilai parameter PID yang didapatkan melalui metode *Tyres-Luyben* adalah  $K_p=4,14$ ,  $K_i=1,727$ , dan  $K_d=0.042$ . pada *setpoint* 30 Rpm memiliki hasil yang lebih baik dengan sistem dinamik sebagai berikut nilai waktu tunda (td) sebesar 1,381 detik, nilai waktu naik ( $t_r=5\%-95\%$ ) sebesar 5,868 detik, nilai waktu naik ( $t_r=10\%-90\%$ ) sebesar 4,379 detik , nilai waktu tunak ( $t_s=5\%$ ) sebesar 5.979 detik, nilai waktu tunak ( $t_s=2\%$ ) sebesar 7,972 detik, nilai waktu tunak ( $t_s=0,5\%$ ) sebesar 9,965 detik dan nilai *error steady state* sebesar 1.59%..

### Saran

Fungsi alih *plant* ini hanya dapat digunakan pada *plant* pada penelitian ini dikarenakan perbedaan nilai pada beberapa parameter. namun peneliti menyarankan untuk menggunakan metode lain seperti *Sliding Mode Control* (SMC), Menggunakan *Genetic Algoritim* (GA), Model *Predictive Control* (MPC) dan lain-lain untuk mendapatkan hasil respon kecepatan yang lebih baik.

## DAFTAR PUSTAKA

- Adriansyah, A., Hidayatama, O. 2013. "Rancang Bangun Prototipe Elevator Menggunakan Microcontroller Arduino AtMega 328P". Jurnal Teknologi Elektro, Universitas Mercu. Vol. 4 (3): pp.100-112
- Alfa Syahroni L, Yosin dan M. Syariffudien Zuhrie. 2018. Rancang Bangun Pengendalian Vibrasi Pada Single Link Flexible Joint Manipulator Robot Berbasis PID Controller. Jurnal Teknik Elektro. Vol 7:3, 183-190.
- Anonymous. 2016. Pengertian Rotary Encoder. (<https://karyainovasiku.blogspot.com/2016/02/pengertian-rotary-encoder.html>) diakses pada 22 Desember 2018.
- Erdon, Jamos. 2012. Pengenalan Sistem Instalasi Elevator(Lift). (<https://erdon.wordpress.com/2012/03/30/-pengenalan-sistem-instalasi-elevator-lift/>). Diakses 26 Febuari 2019
- Khakim, Arif Lukman. 2015. Rancang Bangun Alat Timbang Digital Berbasis Avr Tipe Atmega32. PPs Universitas Negeri Semarang.
- Kho, Dickson. 2017. Pengertian Motor DC Dan Prinsip Kerjanya.([www.teknikelektronika.com/pengertian-motor-dc-prinsip-kerja-dc-motor/](http://www.teknikelektronika.com/pengertian-motor-dc-prinsip-kerja-dc-motor/)) diakses pada 25 Januari 2019
- Ogata, Katsuhito. 2010. Teknik Kontrol Automatik. Terjemahan Edi Leksono. Jakarta: Erlangga.
- Purba, Jon. 19 Maret 2010. Lift Elevator. (<http://jonpurba.wordpress.com/2010/03/19/lift/>).Diakses 16 Januari 2019.
- Sarwono, Jonathan. 2006. Metode Penelitian Kuantitatif dan Kualitatif. Yogyakarta: Graha Ilmu.
- Syahwil, Muhammad. 2017. Panduan Mudah Belajar Arduino Menggunakan Simulasi Proteus + cd. Jakarta : Andi Offset.
- Taan, Ahmad. 2015. PID Controller Tuning. (<https://www.slideshare.net/AhmadTaan/pid-controller-tuning-49463158>) diakses pada tanggal 11 Januari 2019.
- Temel, Sena, dkk. 2013. P, PD, PI, PID CONTROLLERS. Electrical And Electronics Engineering Department, Middle East Technical University.